

(43) Date of A Publication 29.05.2002

(21) Application No 0023952.5

(22) Date of Filing 29.09.2000

(71) Applicant(s)
Reservoirteam Ltd
(Incorporated in the United Kingdom)
43 Alexandra Road, ST ALBANS, Hertfordshire,
AL1 3AU, United Kingdom

(72) Inventor(s)
Joost Christian Herweijer

(74) Agent and/or Address for Service
Reddie & Gross
16 Theobalds Road, LONDON, WC1X 8PL,
United Kingdom

(51) INT CL⁷
G06F 17/60

(52) UK CL (Edition T)
G4A AUB
U1S S1252 S1269

(56) Documents Cited
GB 2313933 A **EP 0314596 A2**
EP 0314594 A2 **WO 00/30000 A2**
WO 00/07125 A1

(58) Field of Search
UK CL (Edition T) **G4A AUB AUBD AUXF AUXX**
INT CL⁷ **G06F 17/00 17/30 17/60**
Online:WPI, EPODOC, JAPIO

(54) Abstract Title
Knowledge based project management system

(57) A project management system has a knowledge module and a display module 8. The knowledge module stores information as a plurality of identifiable units, each unit being logically linked to one or more other units. The display module selectively creates a graphical representation of the units as tree structures. This allows a user to select a given unit of information and select other units which are logically linked to a first unit selected.

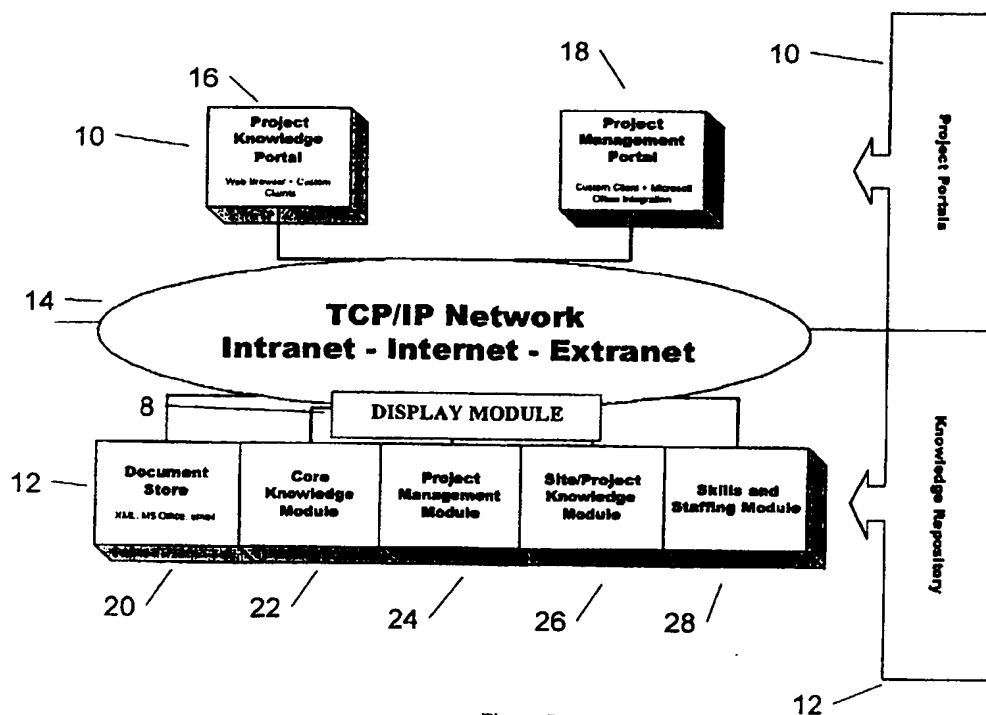


Figure 5

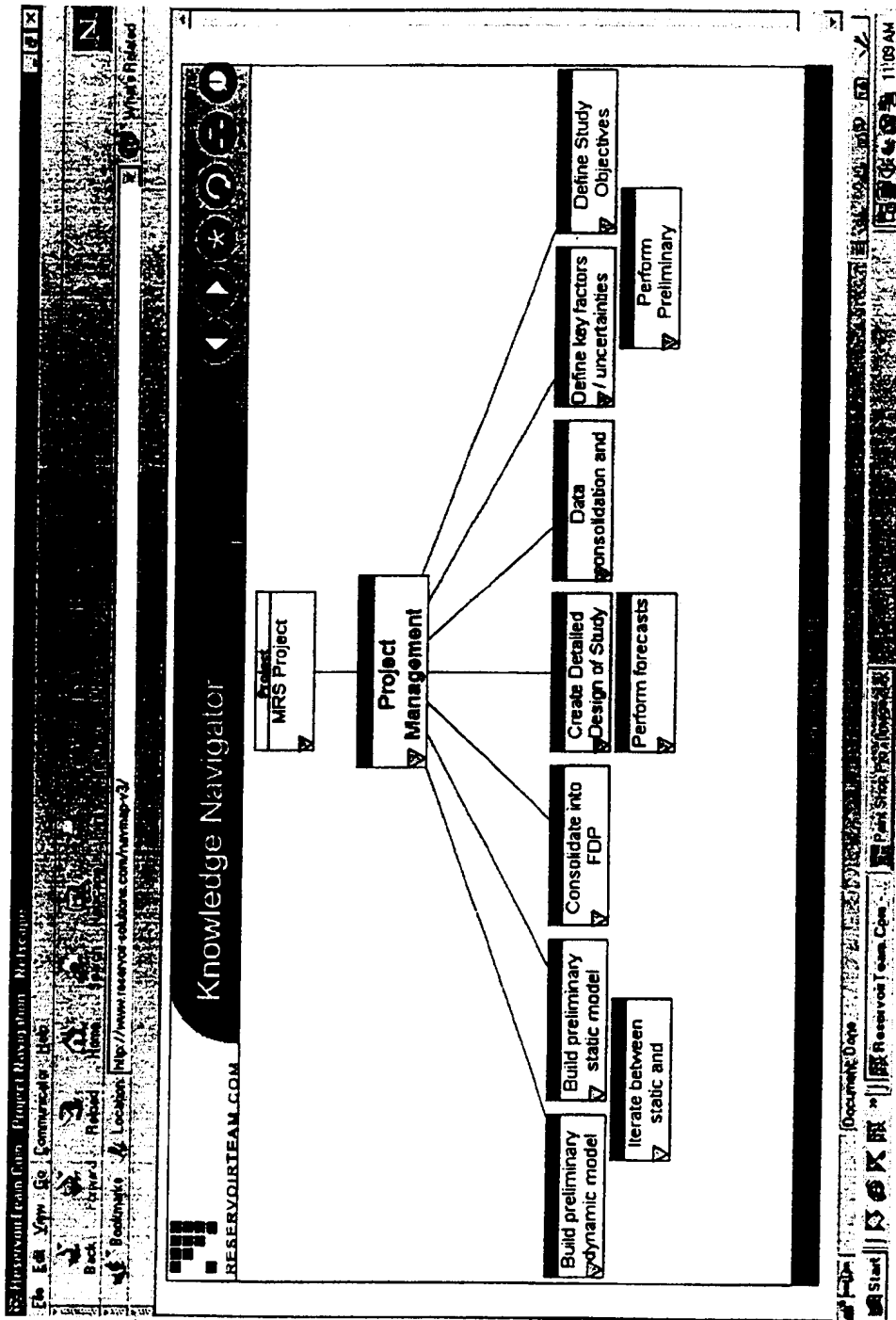


Figure 1

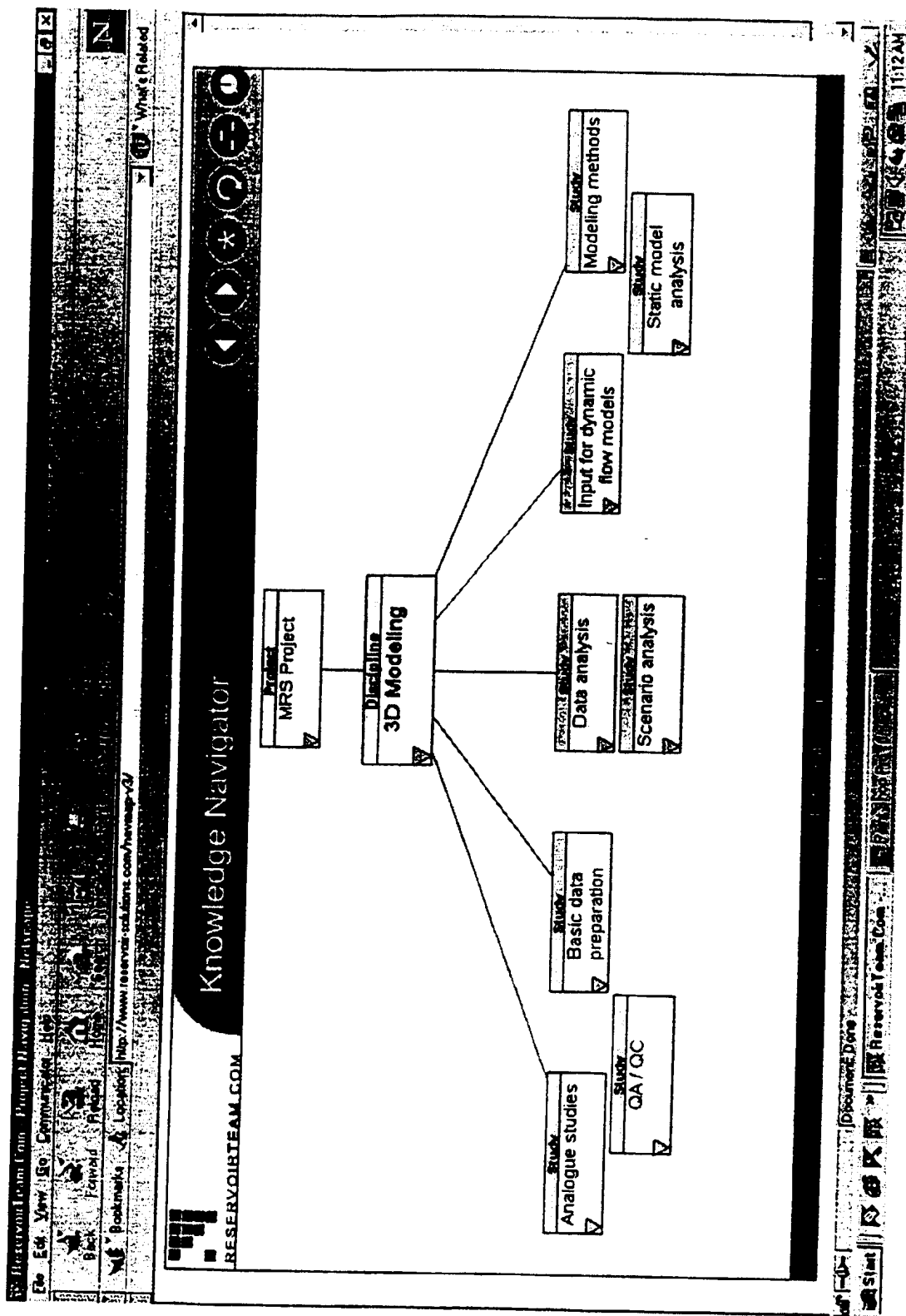


Figure 2

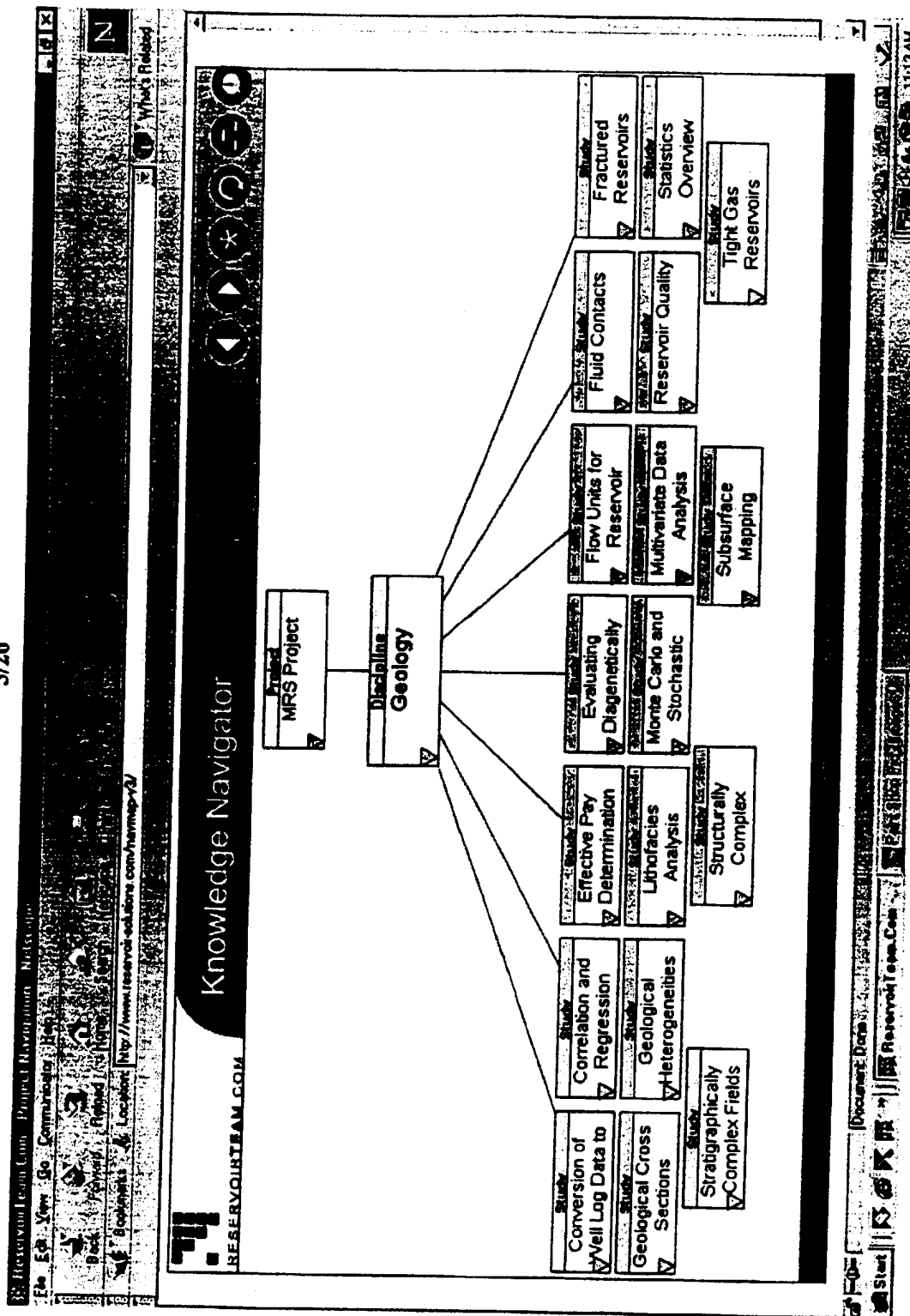


Figure 3

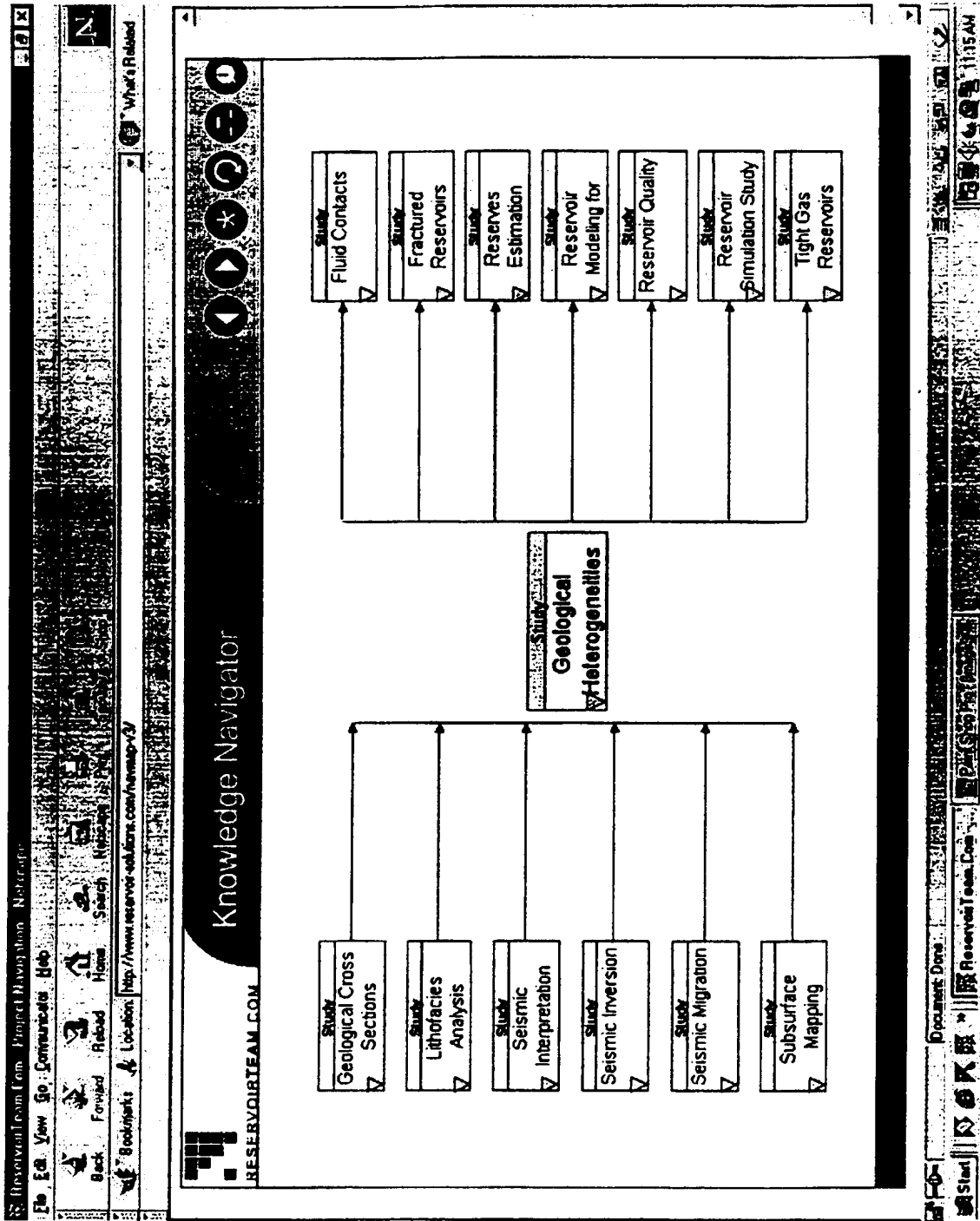


Figure 4

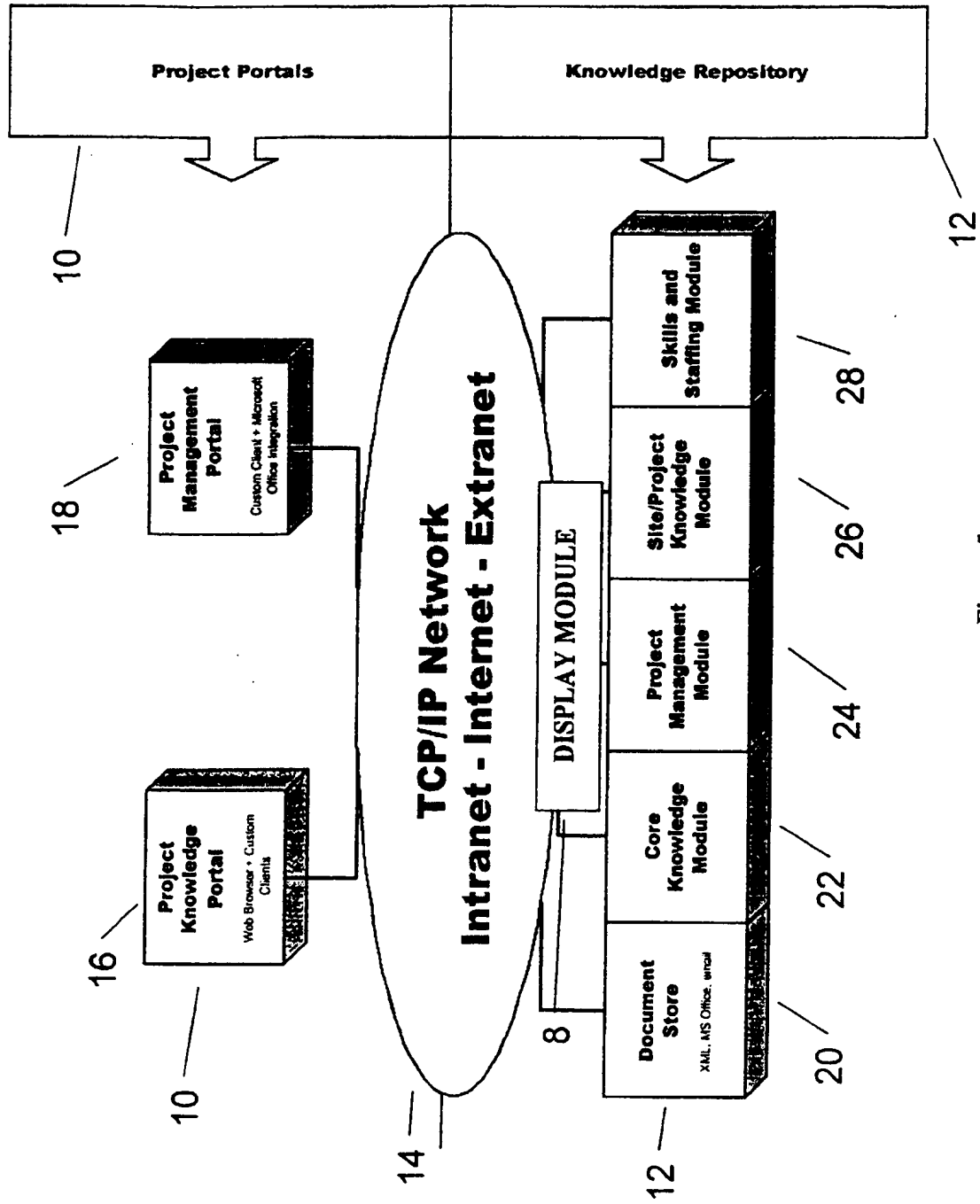


Figure 5

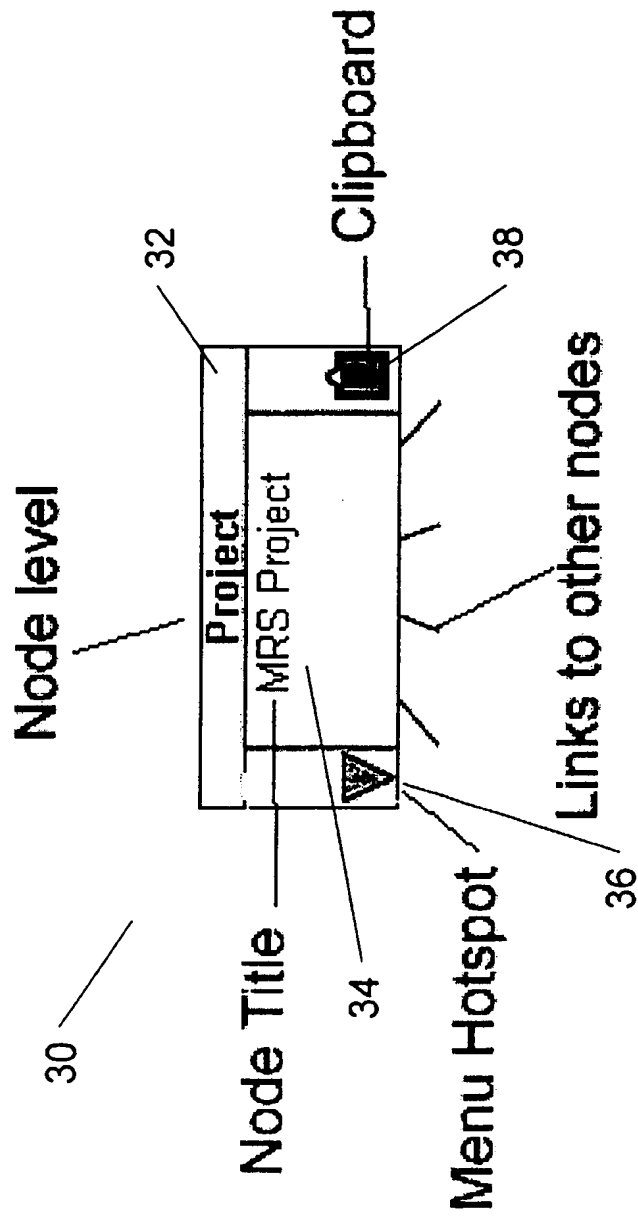


Figure 6

Caption	Description	Seq. #:
MRS Project	The Multidisciplinary Research Study	50

Non-alphabetic characters may cause failure... please report these.

Keywords	Attached	Available
Keyword 1 Keyword 2 Keyword 3 Keyword 4 Keyword 5 Keyword 6 Keyword 7 Keyword 8		

40

X ✓

<input type="button" value="Search for keywords"/>
<input type="button" value="Add new keyword"/>

Keywords retrieved from server 44

42

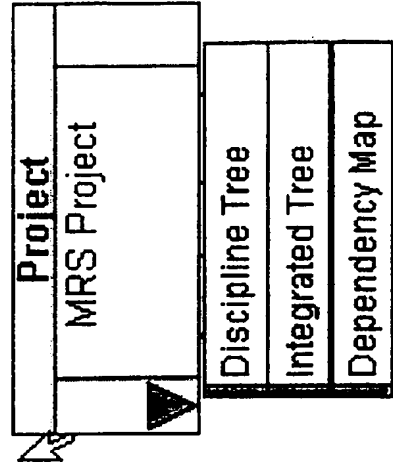
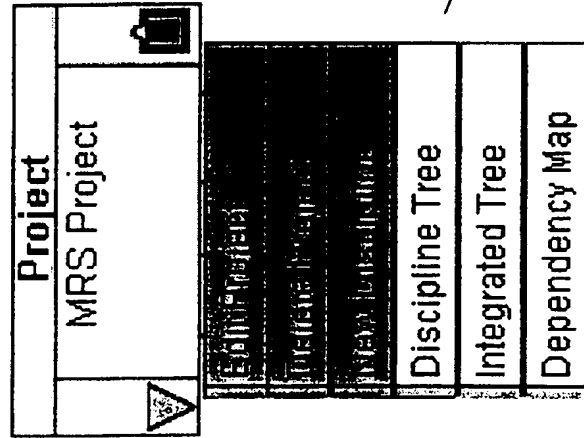
46

8/26

48

Builder Mode

Normal Mode



50

Figure 8

Field	SQL Data Type	Description	
node	varchar(10)	Unique numerical identifier for a Workflow/Knowledge node. Serves as a Primary Key.	50
caption	varchar(50)	Short caption displayed on the node.	52
description	varchar(255)	Longer description. Displays in status line at the base of the screen.	54
level	int(2)	Level of the node. Semantics of the levels are stored in a separate relational table.	56

Field	SQL Data Type	Description	
from_node	varchar(10)	Unique numerical identifier (Primary Key) of the node that is being linked from.	58
to_node	varchar(10)	Unique numerical identifier (Primary Key) of the node that is being linked to.	60
link_mode	enum	Linking mode.	62

Figure 10

Diagram 2: Hypothetical Trees for Discipline D1 and Integrated Project I

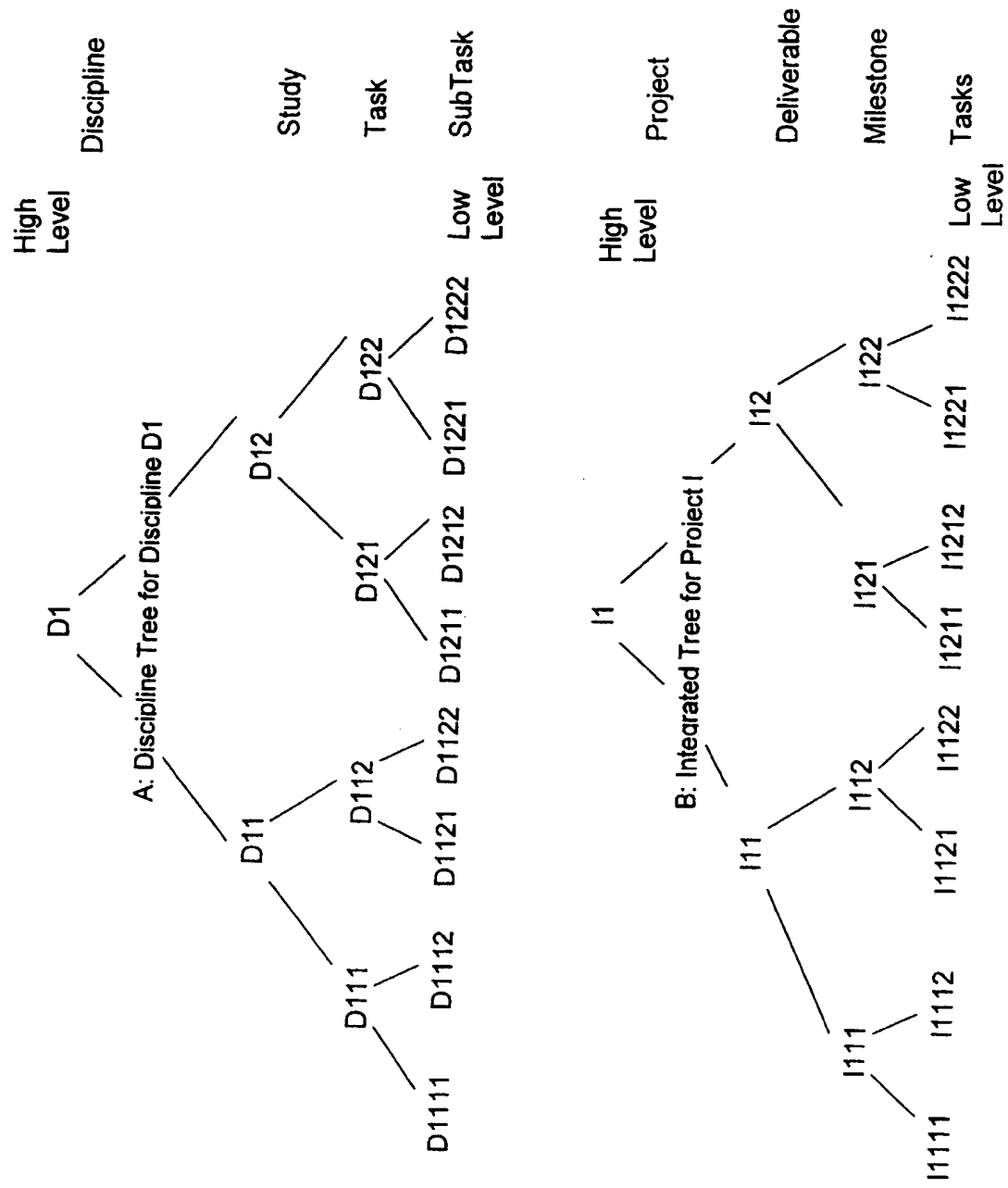


Figure 11

12/26

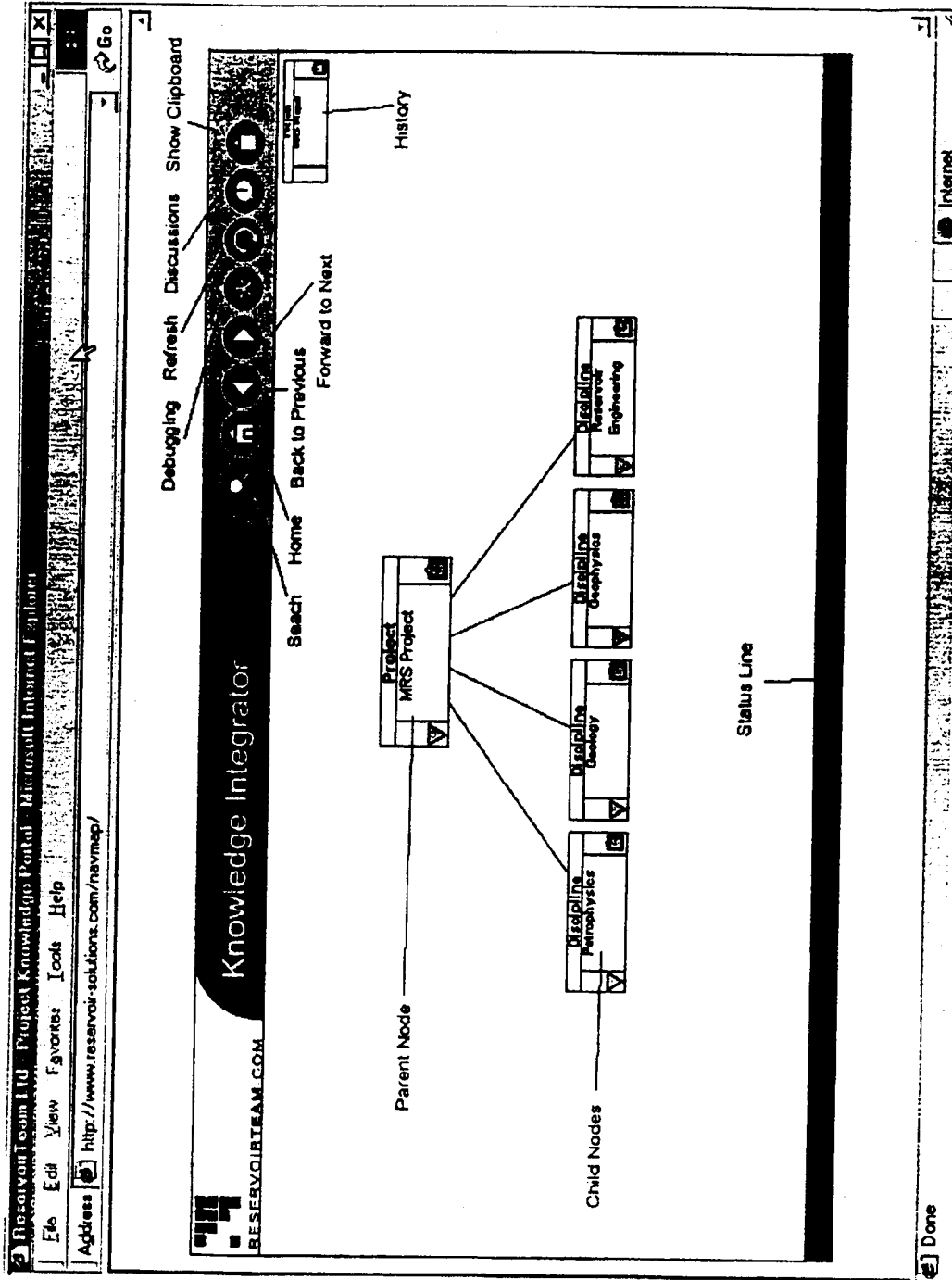


Figure 12



Figure 13

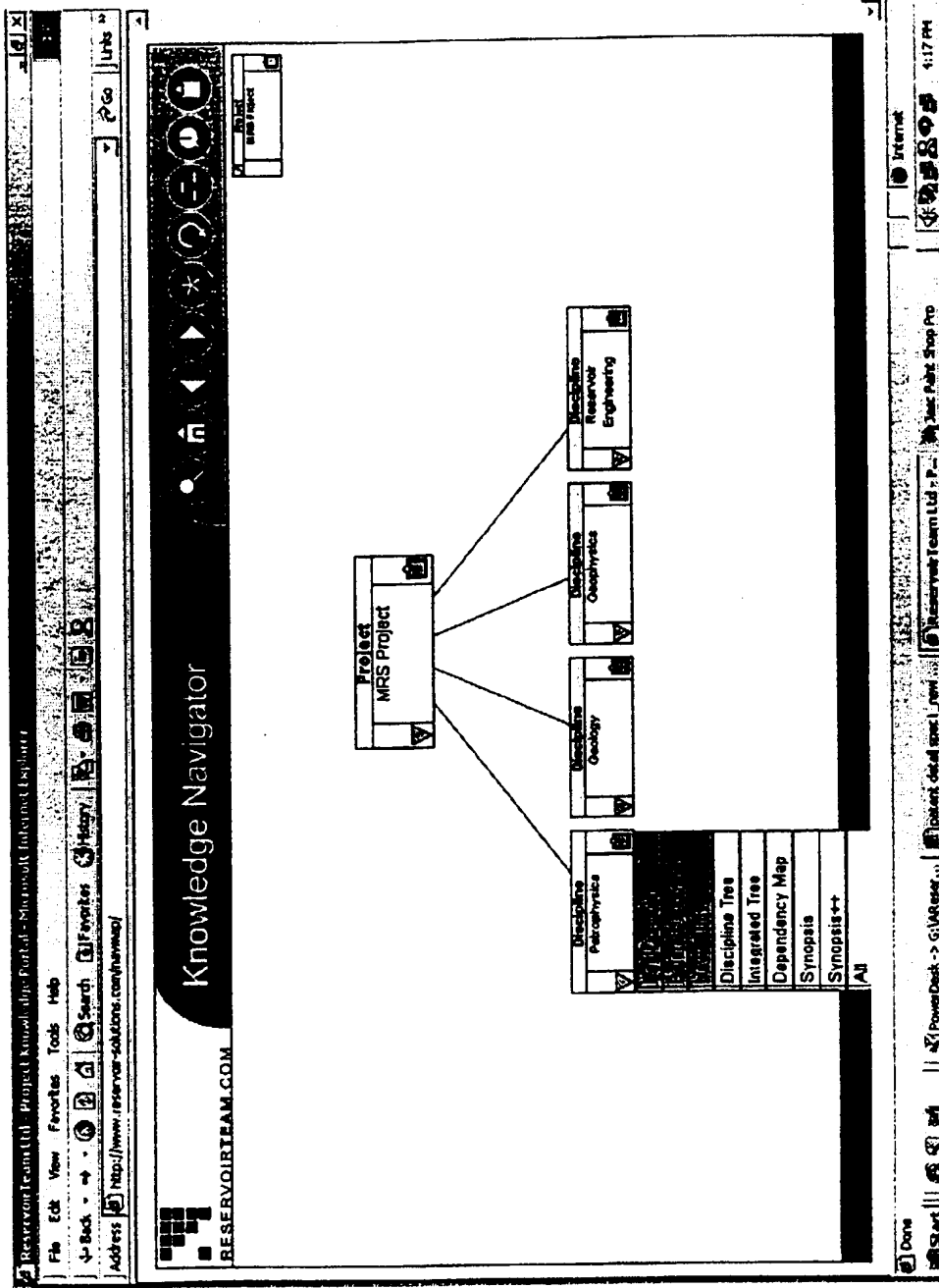


Figure 14

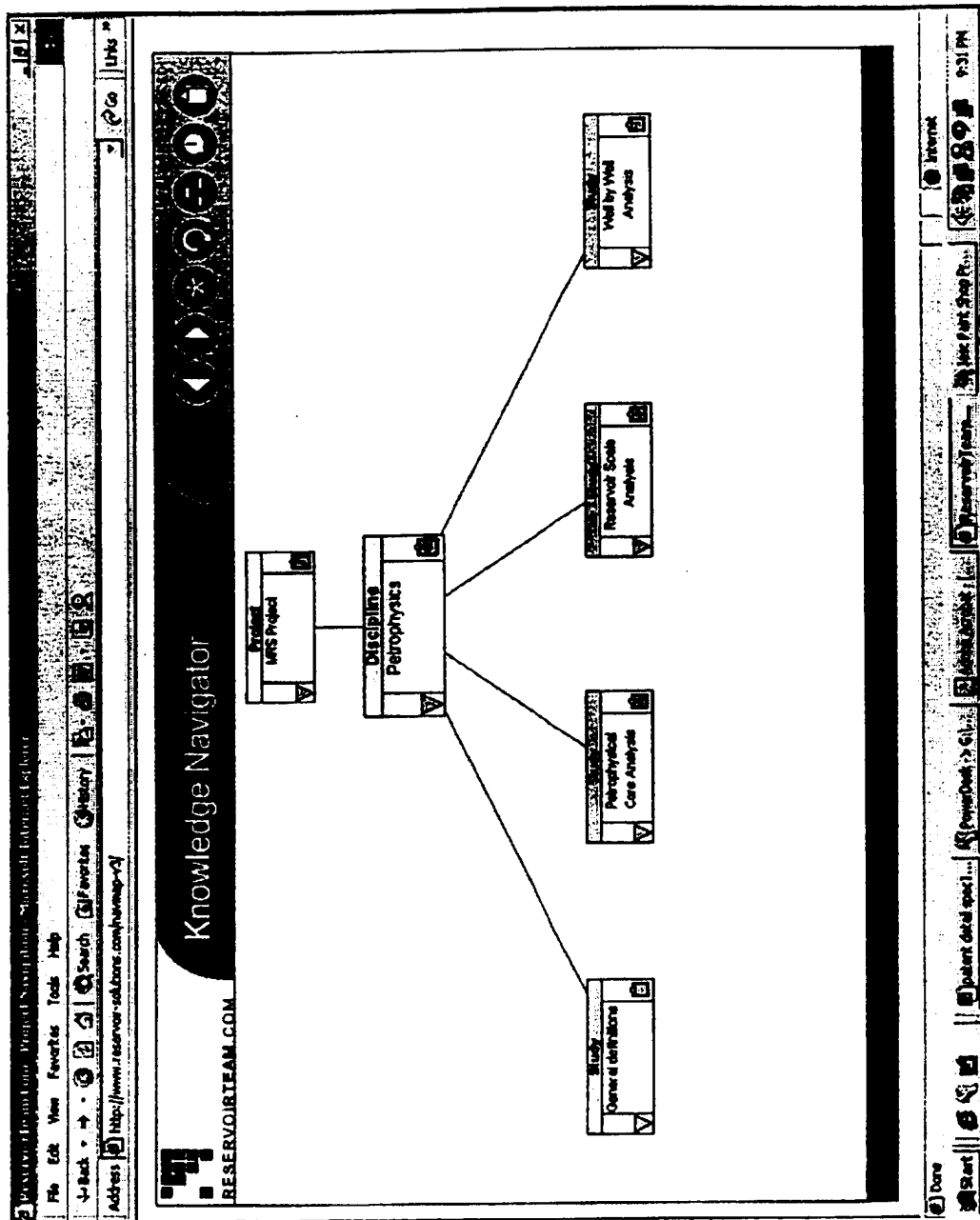


Figure 15

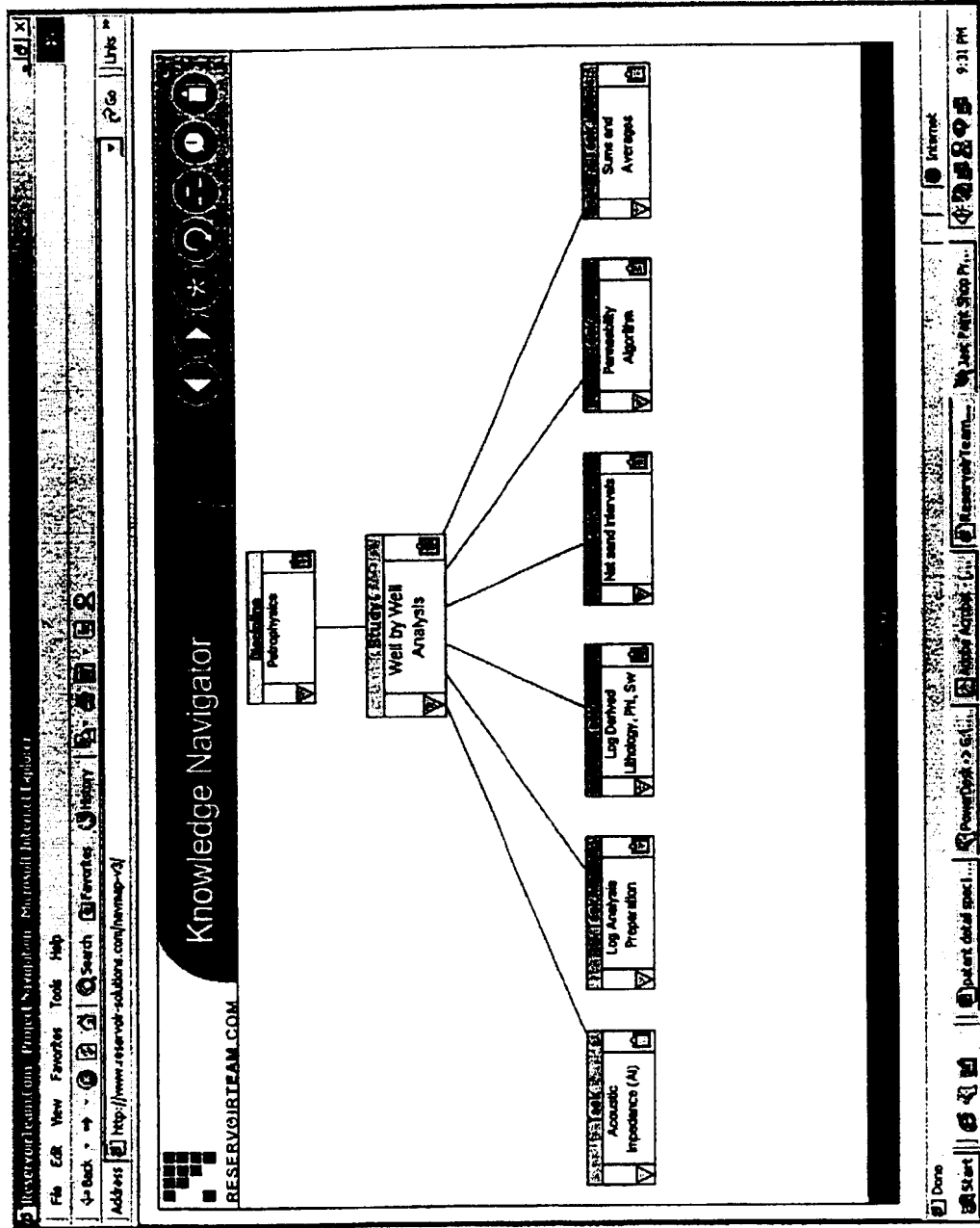


Figure 16

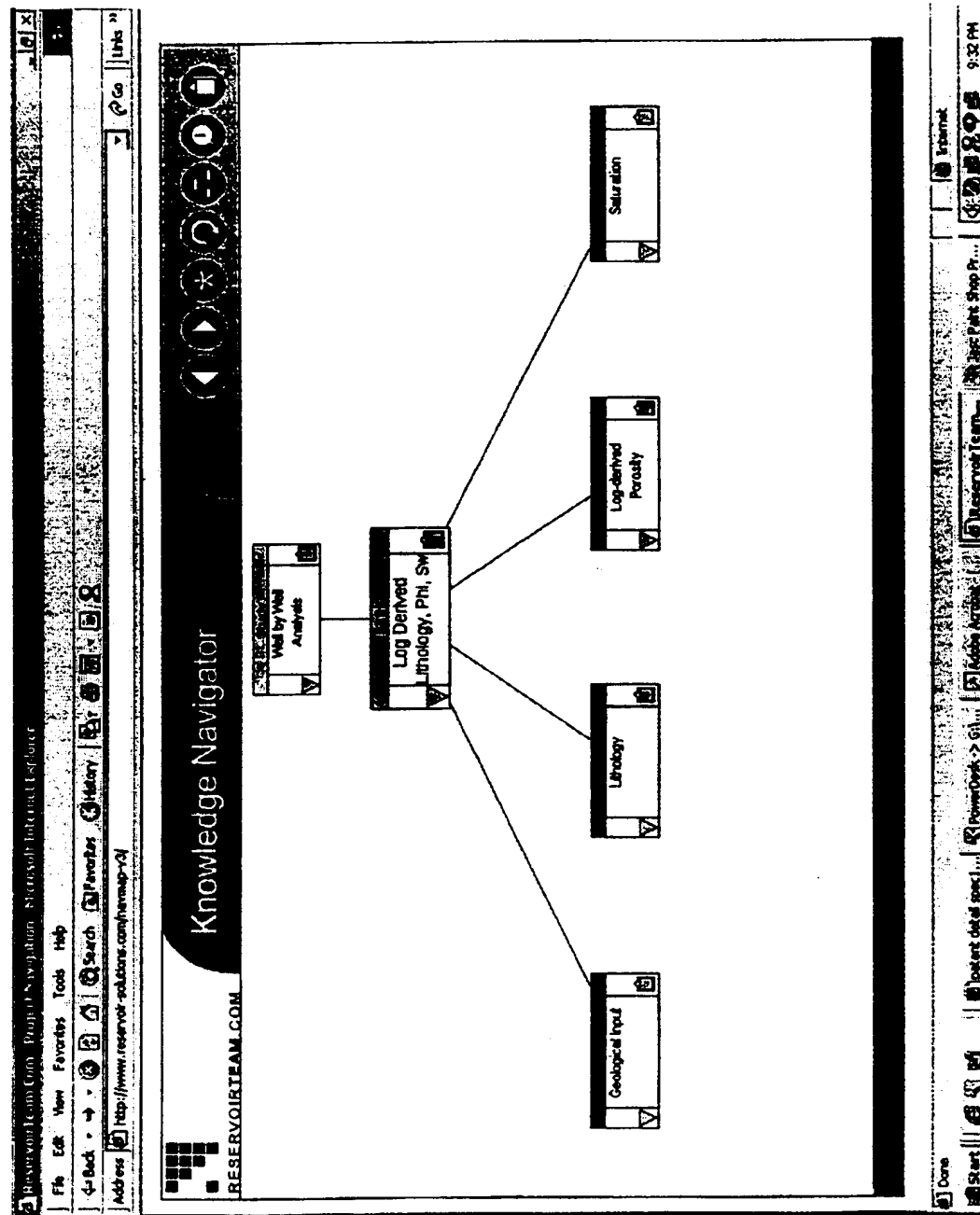


Figure 17

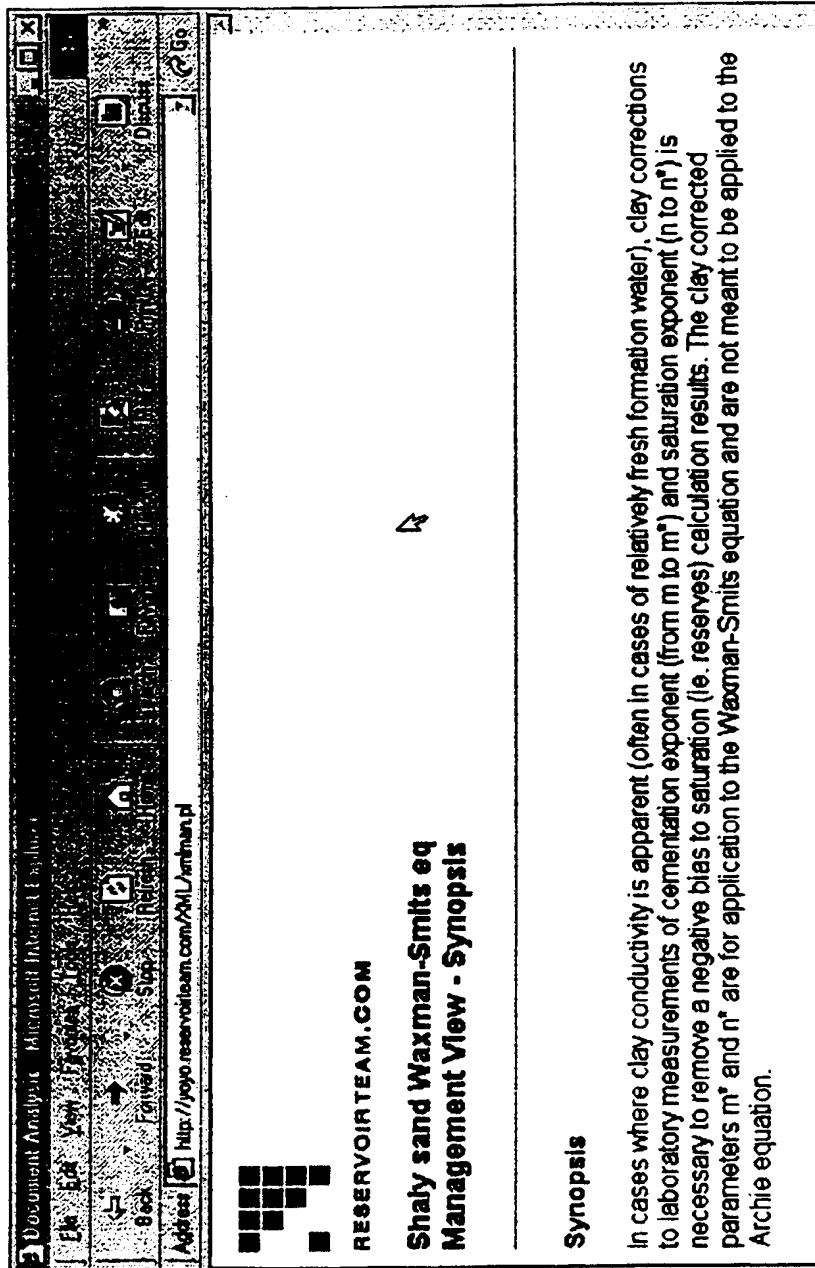


Figure 19

20/26

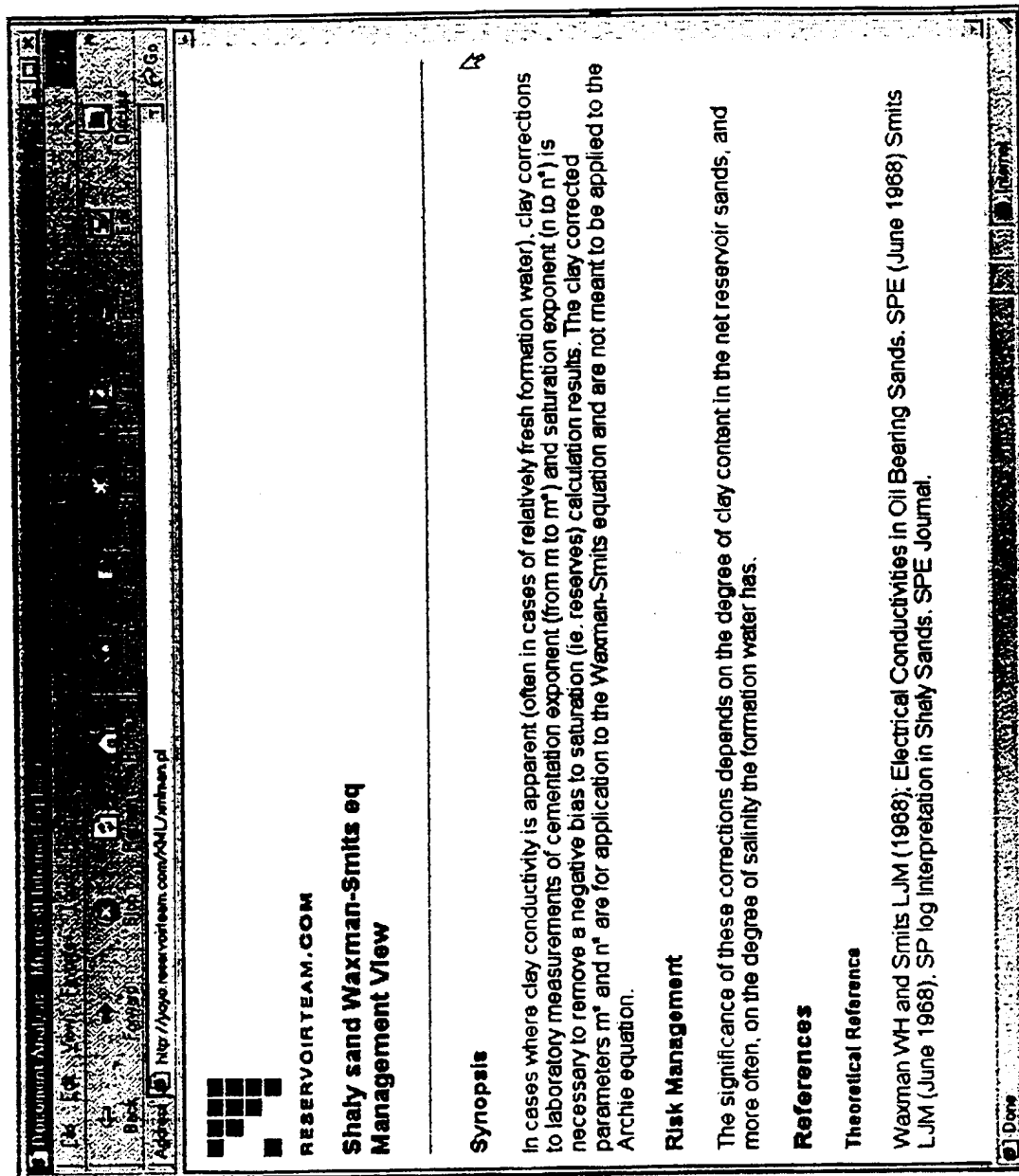


Figure 20

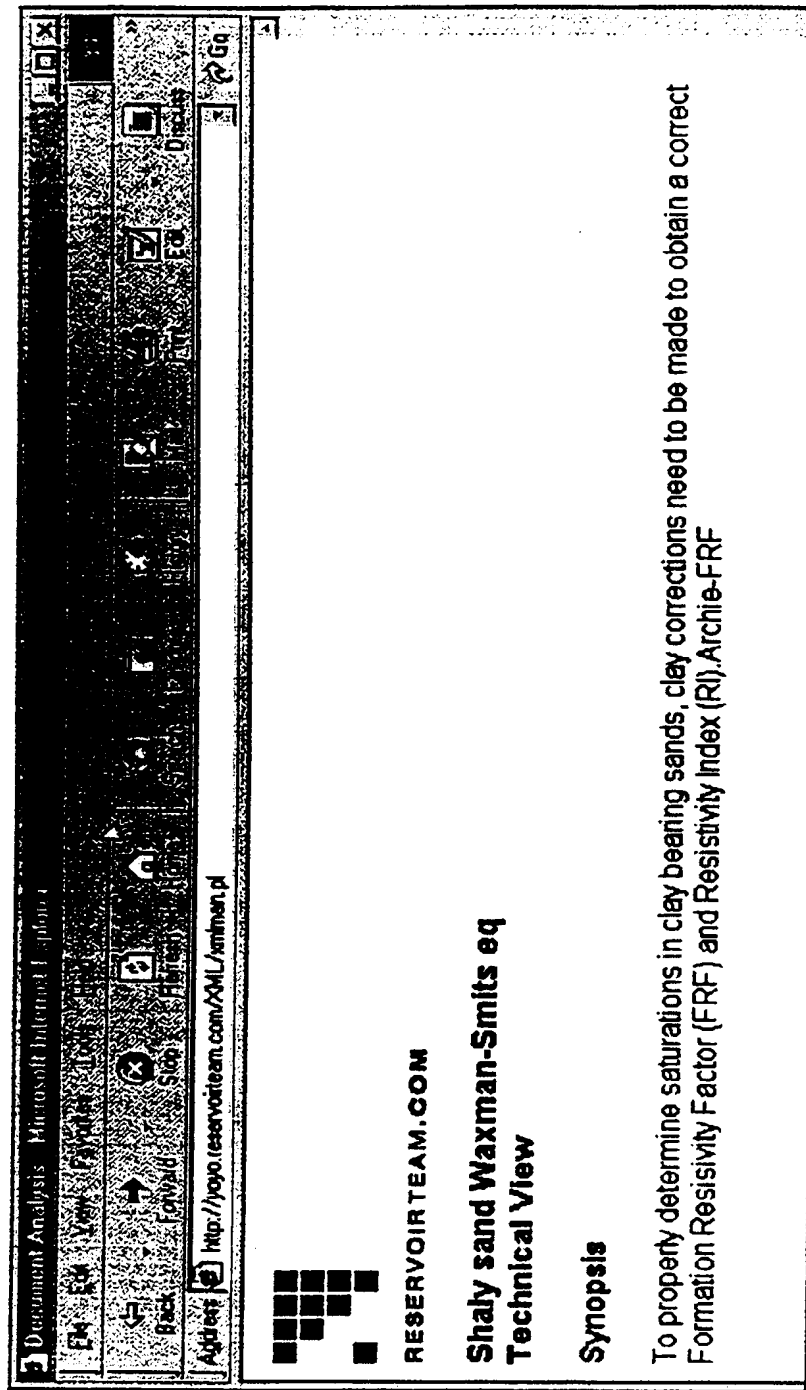


Figure 21

```

<?xml version="1.0"?>
<!-- edited with XML Spy v3.0.7 (http://www.xmlspy.com) by Simon McGuire (Herenow Solutions) -->
<!DOCTYPE nodedoc SYSTEM "M:\XML\DTDs\nodedoc.dtd">
<nodedoc title="Shaly sand Waxman-Smiths eq" author="gb" owner="rlcom">
  <managinfo>
    <synopsis>In cases where clay conductivity is apparent (often in cases of relatively fresh formation water),
    clay corrections to laboratory measurements of cementation exponent (from m to m*) and saturation exponent (n to
    n*) is necessary to remove a negative bias to saturation (ie. reserves) calculation results. The clay corrected
    parameters m* and n* are for application to the Waxman-Smiths equation and are not meant to be applied to the Archie
    equation.
  </synopsis>
  <riskmanag>The significance of these corrections depends on the degree of clay content in the net reservoir
  sands, and more often, on the degree of salinity the formation water has.
  </riskmanag>
  <managinfo>
    <techinfo>
      <synopsis>To properly determine saturations in clay bearing sands, clay corrections need to be made to
      obtain a correct Formation Resistivity Factor (FRF) and Resistivity Index (RI).<link>Archie-FRF</link>
    </synopsis>
    <theory>The clay correction to the measured Formation Resistivity Factor and Resistivity Index accounts for
    the additional conductivity that clay crystals within the sample will contribute. <theory>
    <synopsis>Clay corrections to FRF and RI are performed using respective measurements of rock sample CEC
    (Cation Exchange Capacity), grain density, porosity, and respective FRF or RI. </synopsis>
    <synopsis>The value of B (equivalent conductance of sodium exchange cations) is calculated using the
    laboratory brine resistivity and temperature.</synopsis>
    <procedure>The clay conductivity potential of a rock sample is measured by its Cation-Exchange-Capacity
    (CEC). </procedure>
    <theory>The FRF measurement of clay bearing sands is under-estimated as a result of the increased bulk
    conductivity (contributed by clay crystals). Likewise occurs in RI measurements. The purpose of the below detailed
    clay-corrections is to correct for these under-estimates.
  </theory>
  <procedure>The equivalent conductance of sodium exchange cations (B) is determined from the following
  empirically derived relation from Waxman and Thomas. When determining B for the purpose of performing the clay
  correction on lab measurements, one must employ laboratory conditions of brine resistivity and temperature.

```

Figure 22

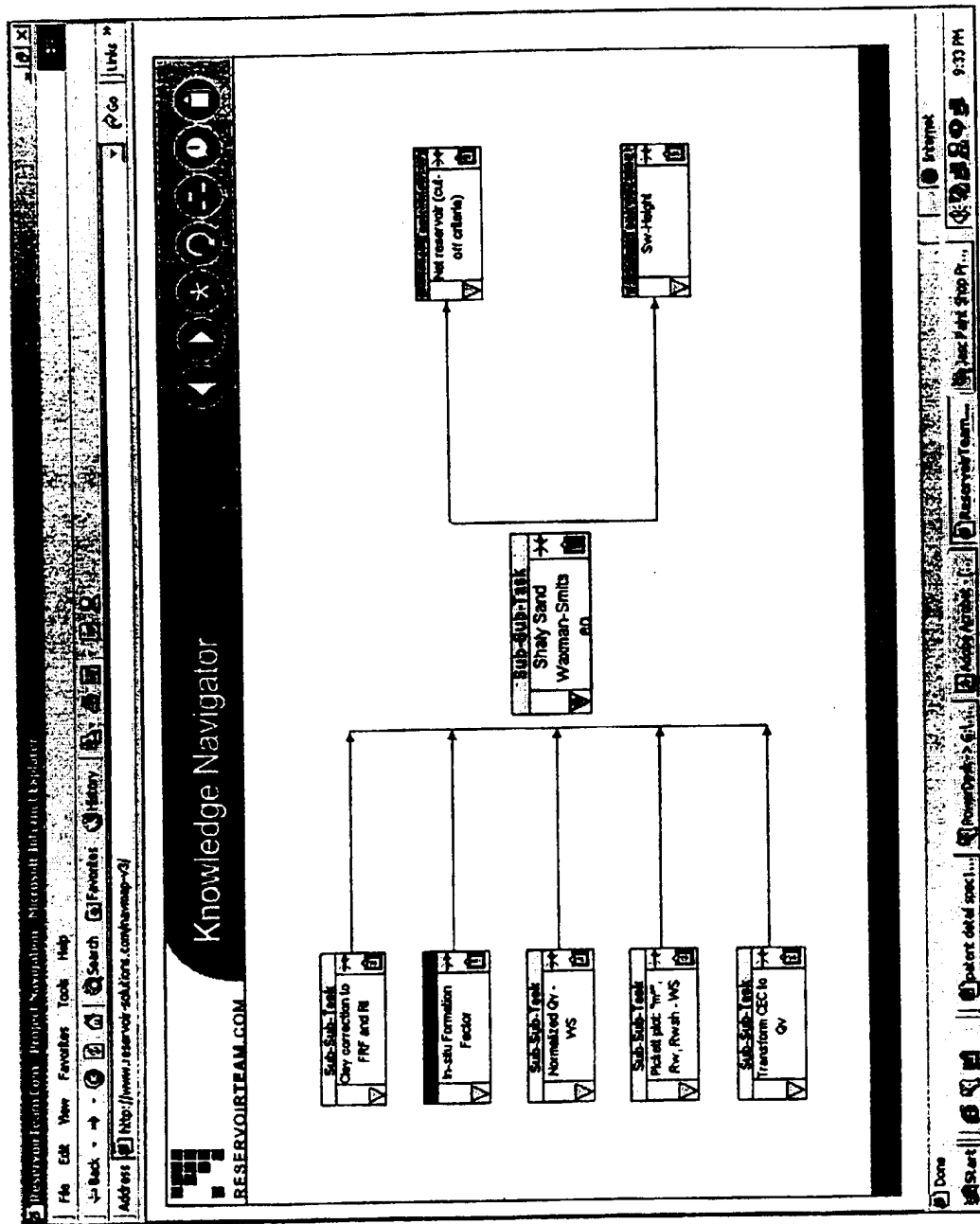


Figure 23

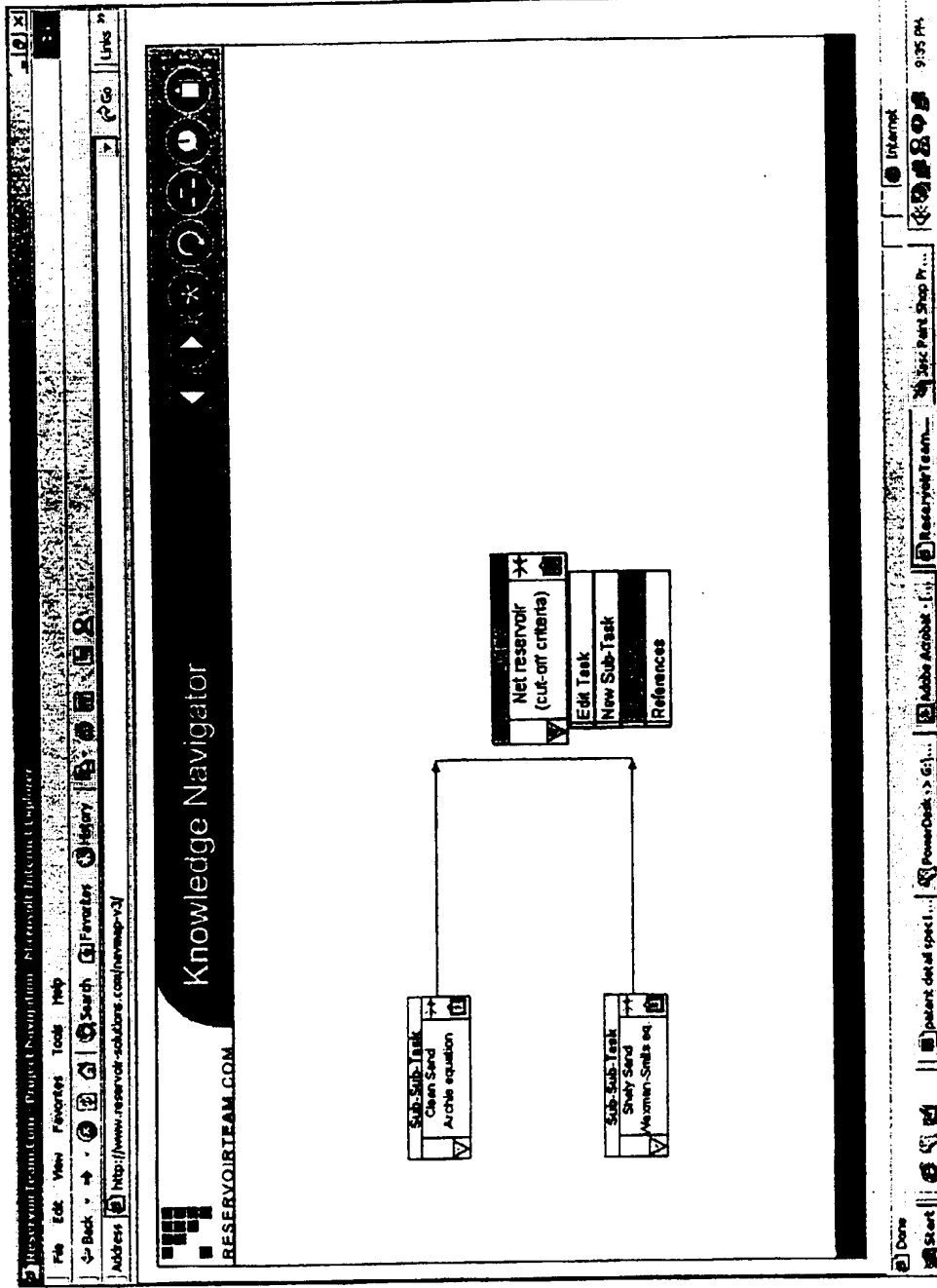


Figure 24

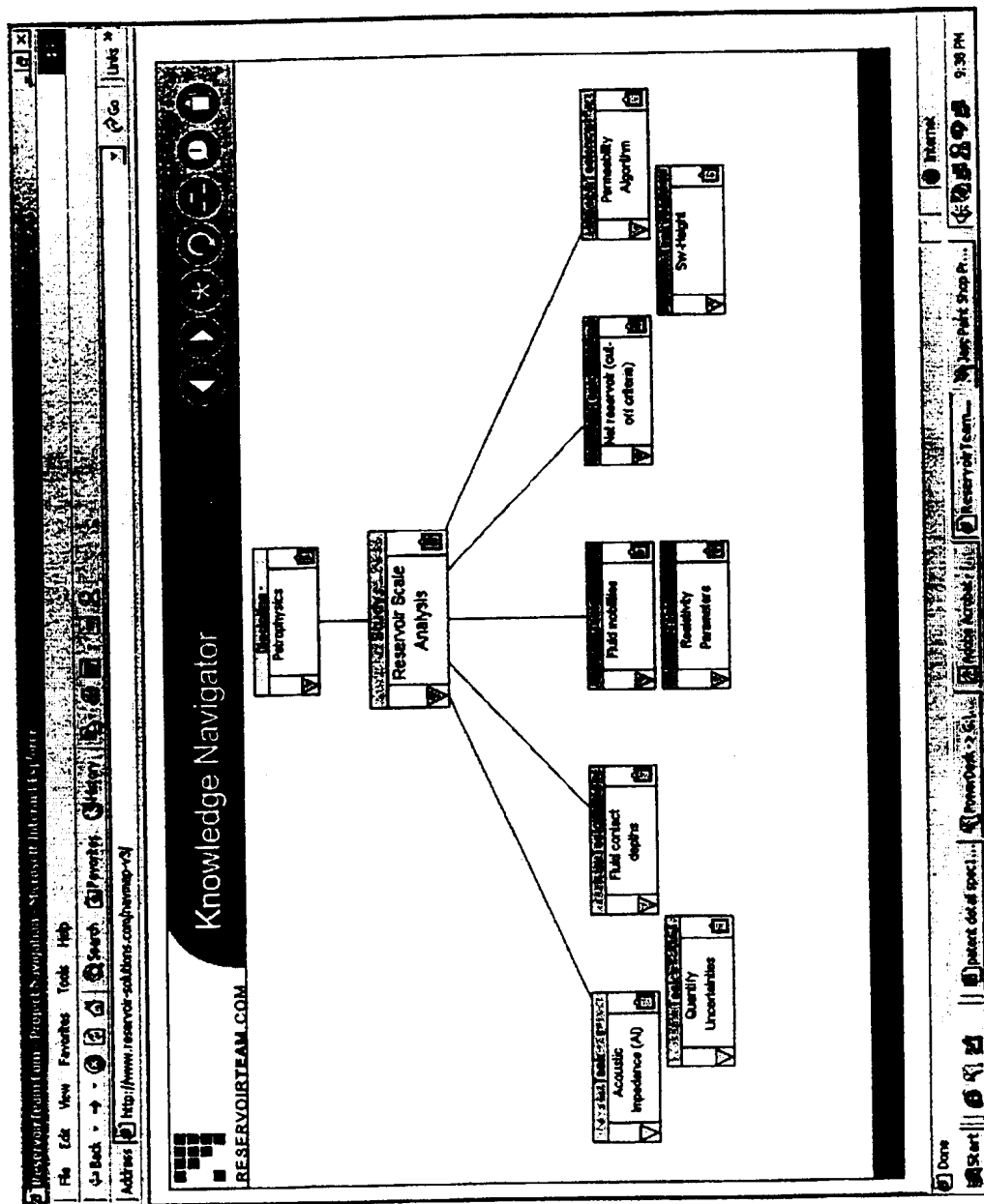


Figure 25

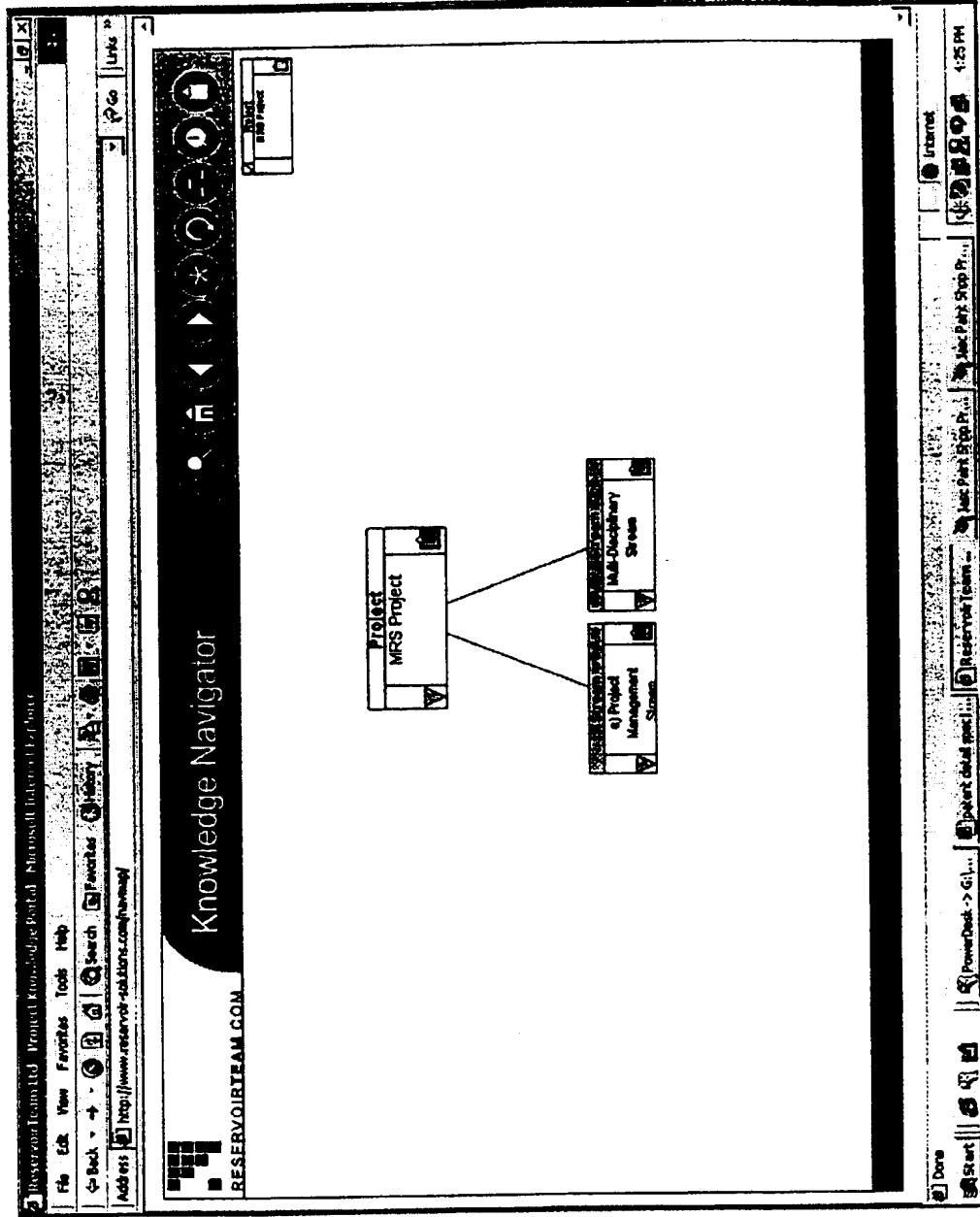


Figure 26

PROJECT MANAGEMENT SYSTEM AND METHODField of the Invention

The present invention relates to methods and systems for project management. In particular, an Internet/intranet based multi-dimensional knowledge database for multi-disciplinary projects combined with a visual tool to map/organize, navigate, and search this database and create a multi-dimensional project map is provided.

Background to the Invention

Multi-disciplinary studies combine a variety of workflow processes, individual pieces of knowledge, and interpretations. All these are typically "owned" by specialists and are often worked on and reported on in isolation and ultimately managed/collated in a non-systematic way (for example by a supervisor or project manager who has "the overlook"). Project management consists of creating timelines and hence leads to a one-dimensional (task-time) overview of the project. No systematic overview is maintained of the linkage, across disciplines, between the different workflow processes, individual pieces of knowledge and various interpretations. The result is a large dependency on the capability of the project manager to understand and coordinate a project, usually involving a lot of face to face communication, which requires very frequent meetings and physical co-location.

There have been attempts to create resource and management systems using computer networks. Systems such as Microsoft Project ™ and Lotus Notes ™ are software packages which typically run on an Intranet network. These software packages facilitate storage and retrieval

of documents to allow team collaboration. However, they are limited in their usefulness to full project management.

5 The World Wide Web is itself a further example of a collaboration system. HTML was written to allow users to collaborate across large distances by publishing text in a standard format. In addition, those publishing text can embed links to other documents stored at other computers or networks to allow navigation. However, as users of the Internet know, the Web is unstructured leading to
10 difficulties in finding documents. Search engines were developed to alleviate this problem. However the Web cannot, itself, facilitate project management.

15 We have appreciated the need for a system to assist in project management. In particular, we have appreciated that networks such as Intranets/the Internet can be enhanced to assist project management.

20 We have further appreciated the need to coherently manage teamwork, and to provide coherent structures that combine data, technical knowledge and managerial know-how for future projects. We have also identified the need to minimise the effect of exchanging given specialists on a project for other specialists.

Summary of the Invention

25 Accordingly, in a broad aspect, the invention provides a project management system and method in which data is stored in a structured manner.

30 In particular, there is provided a project management system, comprising a knowledge module arranged to store information relating to a project as a plurality of

identifiable units, each identifiable unit being logically
linked to one or more other identifiable units; a display
module arranged to selectively create a graphical
representation of the identifiable units as a tree
5 structure of multiple possible tree structures, the tree
structure showing the links between each of the units; and
wherein the knowledge module and display module are
further arranged to retrieve, on user selection of a given
unit, further units being logically linked to the given
10 unit and to create a further graphical representation of
those further units.

An advantage provided by a system embodying the invention
is that users can view a project in variety of ways by
selecting a view of different parts of the project from
15 the same underlying data. Thus multiple alternative tree
structures are available.

The invention and preferred features are set out in the
claims to which reference is now directed.

An embodiment of the invention is an Internet/Intranet
20 database of identifiable units of information known as
workflow nodes, for which knowledge entities are stored,
such as: workflow process description, interpretation
techniques, available data, specific interpretations,
management information such as time estimates, critical
25 path issues, and skill sets of staff capable to work on
the specific process. An interface is provided to create
multiple streams to connect these nodes, for example
specialist discipline hierarchical trees, process oriented
hierarchical trees, and process oriented sequences. The
30 user-interface can also create links from the one tree
into the other tree. As such a dynamic (multi-dimensional)
project map is created.

Brief Description of the Figures

An embodiment of the invention will now be described, by way of example only, and with reference to the figures, in which:

- 5 Figure 1: is a screen view of project management aspects;
- Figure 2: is a screen view of a Knowledge tree;
- Figure 3: is a further screen view of a knowledge tree;
- Figure 4: is a screen view of processes;
- 10 Figure 5: is a functional diagram of the main components of the system embodying the invention;
- Figure 6: is a schematic representation of a workflow/knowledge node;
- Figure 7: is a schematic view of a screen allowing a war
- 15 Figure 8: is a further view of the mode shown in Figure 7;
- Figure 9: is a representation of storage of nodes in a relational database;
- 20 Figure 10: is a representation of storage of links between nodes in a relational database;
- Figure 11: is a representation of a discipline tree;
- Figure 12: shows an interface;
- Figure 13: shows the graphical interface opening screen;
- 25 Figure 14: shows a discipline tree;
- Figure 15: shows a further level in a discipline tree;
- Figure 16: shows a further level in a discipline tree;
- Figure 17: shows a further level in a discipline tree;
- Figure 18: shows a sub-sub task in a discipline tree;
- 30 Figure 19: shows an information display;
- Figure 20: shows an information display;
- Figure 21: shows an information display;
- Figure 22: shows an information display;
- Figure 23: shows a dependency map;

Figure 24: shows navigation using a dependency map;
Figure 25: shows a discipline tree from Figure 24; and
Figure 26: shows the top level of a tree.

Description of An Embodiment

5 The embodiment of the invention is Knowledge Management
software driven by a dynamic project map that provides the
first unified work environment for the three major aspects
of complex studies: technical training and real-time
guidance, workflow optimisation and project management,
10 and knowledge capture and results reporting.

The dynamic project map is described with respect to the
petroleum industry and allows discipline-specialist
members and managers of multi disciplinary teams to
systematically visualise, (re-)design, and execute the
15 complex interdependent work flows of multi disciplinary
projects, as well as store, update, and retrieve the
resulting technical, workflow, and business knowledge.
The technology is underpinned by industry-specialist
knowledge organised by hot-linked workflow nodes that can
20 be viewed in either a Project Management Stream, a Multi-
Disciplinary Stream, a Discipline Specialist Stream, or a
"Process View" that shows the interdependencies between
them. Whilst described with reference to the upstream
petroleum (exploration and production) industry, the
25 invention can be applied to all areas of technology where
different specialist technology disciplines are working in
parallel towards solving complex problems and where the end
result is highly dependent. The following areas of
technology have been considered large infrastructure
30 projects, environmental impact assessment studies, and
mine development/construction.

An overview of software embodying the Invention is shown
in Figures 1 to 4. Knowledge is stored and accessed via

software that runs from an Intranet server in a standard browser. The system is based on the concept of workflow nodes (studies, tasks, sub-tasks, etc.) that are graphically depicted in their real-life study sequence.

5 These are units of information relating to a given project. A tree structure is also available that depicts these workflow nodes in a more traditional discipline-oriented hierarchy. At each workflow node a variety of information objects can be stored, such as technical

10 literature, best practice procedures, known pitfalls, tips and hints, miscellaneous reports, e-mails, project specific progresses, references to in-house and external specialists, timing estimates, etc. Keywords are assigned to find and link specific workflow nodes, and are used to

15 illustrate inter-dependencies.

The technology includes comprehensive visualisation that allows several flexible views on the knowledge base described above. Figure 1 is a view of the Project Management System (PMS), Figure 2 is a view of the

20 knowledge tree for 3D modelling, and Figure 3 is a view of the knowledge tree for the discipline geology and shows a set of studies for which tasks, sub-tasks, and sub-sub-tasks can be defined. Figure 4 is a process view, showing the dependencies for a given study from the geology

25 discipline tree. The same knowledge database is accessed, to present these screen views providing a visualisation tool which is interactive, and allows a user to edit the workflow schema, the tree structure, and other links between the workflow nodes.

30 The underlying technology is largely based on Internet protocols, a SQL database centrally stored on a corporate Intranet web server, and XML (eXtended Markup Language). Basic access to the system can be obtained via a standard web-browser, which guarantees independence of a specific

35 computer platform. The entire system can be securely

deployed on a corporate Intranet. Basic documents can be created with standard tools such as MS Word, Excel, Lotus Notes, e-mails, etc. Relevant information can be exported easily to MS Office, MS Word, Excel, Adobe PDF files, HTML web pages, etc. Hence, a tremendous flexibility is achieved for team cooperation across all corporate entities.

The main functional components of a system embodying the invention are shown in Figure 5. The system comprises software resident at a client computer known as project portals 10 and software and data at a server forming a knowledge repository 12. The communication between client and server is by any network 14 such as the Internet or an Intranet/Extranet. The client software 10 comprises a project knowledge portal 16 and/or a project management portal 18. These are communications software programmed in object oriented languages such as Macromedia Flash for web browsers or Visual Basic for Windows applications.

A display module 8 receives data from each of the server side components and creates a graphical view of the information. The display module 8 consists of a server sub-module and a client interface sub-module which communicate with one another. Although shown at the server, the display module could be entirely resident at the server or at the client.

Alternatives thus include having the data transmitted from the server and the graphical view created entirely at the client, or to create the graphical view at the server and transmit this to the client. The preferred choice with the display module makes best use of the limited bandwidth of the Internet.

The client interface sub-module forms part of the client software 10 and accepts instructions as to which project

management data (nodes, links and attributes) and document store content are requested for display and which form of display is required. These instructions are transferred via the TC/IP network (Internet / Intranet) to the server.

5 Upon receipt of the instructions the server sub-module will query the database to retrieve the data necessary (nodes, links, and attributes) and create input instructions for the client sub-module to build the dependency map / tree and populate it with information.

10 The input instructions are transferred to the client and used by the client sub-module to create the visual representation of the dependency map / tree. In the embodiment, these operations are conducted using the Macromedia Flash programming language which allows

15 creation of graphics and the user interface within a standard web browser.

The server side applications 12 form the core of the embodying system, and are based on relational (SQL) databases and XML documents. The basic functional

20 components storing knowledge known as "Workflow Nodes" are stored in the core knowledge Module 22. A user can access the core knowledge module database 22 using the project knowledge portal 16. The basic project management data are stored in the project management module 24, and can be

25 accessed by the project management portal 18.

The knowledge repository 12 also comprises a document store 20 within which documents relating to workflow are stored and a project knowledge module 26 and skills and staffing module 28. The key component is the core

30 knowledge module 22 and the workflow/knowledge nodes stored therein.

Workflow/Knowledge nodes represent the basic subdivision of a project. Knowledge is captured, stored and retrieved

via these nodes. Nodes have 'intelligence' and can present themselves in a different manner depending on the type of user and their privileges. (technical, managerial etc.) The nodes can be logically arranged as trees and
5 dependency maps. A collection of Workflow/Knowledge nodes can assume various structures due to links that exist between nodes that define how a node relates to another node. Nodes can be arranged in a tree structure wherein a parent node can have a number of child nodes and these
10 nodes in turn can have children nodes. One particular tree based structure is the so-called discipline structure, where the project is broken down in a hierarchical structure that reflects a scientific progression of subdividing knowledge within a particular discipline. This
15 structure can also be thought of as an 'expertise taxonomy'.

Another structure is the so-called process mode. For a given Workflow/Knowledge node, dependencies are defined showing how Workflow/Knowledge nodes follow each other in
20 the logical sequence of project workflow. This structure can also be called a dependency map.

Generally, all structures between Workflow/Knowledge nodes can be considered as 'project knowledge road maps'. These provide the user with a dynamic, polymorphous interface to
25 a project management system.

As previously explained, the basic knowledge entities (Workflow/Knowledge nodes and their attributes, links, dependencies) are stored in the core knowledge module database 22. The client application (Project Knowledge
30 Portal 16) interactively accesses this database to build the various dependency maps and knowledge roadmaps. The Project Knowledge Portal also allows the user to interactively create links and dependencies by altering this database.

The basic project management data are stored in the project management server module database 24, and accessed via the client application (Project Management Portal 18). The project management server module also has database
5 links to the core knowledge server module.

The document store 20 and the skills and staffing module 28 are XML documents. XML stands for eXtensible Markup Language and is recent standard developed by the World Wide Web Consortium to replace HTML. XML provides the
10 ability to mark up a document based purely on its content rather than on the way it should be formatted and presented. This means that documents can be processed, queried and filtered based on their informational context. This is not possible with HTML since document markup
15 consists of formatting codes only. Via a workflow/Knowledge node, documents can be easily attached. This allows the interface to find all relevant documents for a given Workflow/Knowledge node (or set of Workflow/Knowledge nodes) and to conduct a query and
20 select information for display or delivery.

Workflow/Knowledge nodes are the preferred form of identifiable units and are the basic subdivision of a project and will now be described. Knowledge is captured, stored and retrieved via Workflow/Knowledge nodes. Nodes
25 have 'intelligence' and can present themselves and their actions in a different manner depending on the type of user and their privileges. (technical, managerial etc). User accounts and their associated privileges are maintained using a username/password system based on the
30 basic NCSA HTTP/1.0 Basic Authentication scheme. Nodes are arranged in different hierarchies so that a user can drill down into specifics from higher level viewpoints. A basic Workflow/Knowledge node 30 and its constituent parts, as represented graphically in the system, are
35 illustrated in Figure 6.

The Node Level 32 is the level of the particular Workflow/Knowledge node in the hierarchy. The Node Title 34 is the title of the particular Workflow/Knowledge node. When the Menu Hotspot 36 is activated by the cursor, a
5 menu of actions appears above or below the node - depending on its location on the screen. The options presented will be sensitive to the context of the user and their privileges. The Clipboard icon 38 appears when the user has the privilege to build dependencies between nodes
10 and primarily serves as a node selection mechanism. When clicked upon, the particular Workflow/Knowledge node is stored in a temporary location so that it can be later selected as a linked node in different context. This is explained further below.

15 When the cursor is held over a Workflow/Knowledge node, an expanded description of the particular node appears in a status line at the base of the screen. When the cursor is held over the same node for more than 2 seconds, a list of the nodes' children replaces the description at the base
20 of the screen.

When the node itself is clicked upon in any region apart from the Menu Hotspot 36 or the Clipboard, the user moves down the next level in the hierarchy and this node appears as the parent node.

25 Workflow/Knowledge nodes 30 also have keywords attached to them. The keywords are sourced from a master list that is made available for a particular project domain. By definition the root project node always contains the superset of keywords. If a particular user has the
30 required privileges for 'a Builder Mode', they may edit or delete attached keywords. The interface for attaching keywords is illustrated in Figure 7. Note that is for the root node so all keywords are attached and none are available.

The screen view of Figure 7 allows keywords 40 to be created and attached. The attachment list 42 shows the attachment and an availability list 44 shows availability. The purpose of the attachment list is to present a predefined list of available keywords to users, rather than allowing any user to create a new keyword which could become confusing. The availability list thus limited use of keywords.

Workflow/Knowledge nodes are always created as children of an existing node. If a particular user has the required privileges for 'Builder Mode', they may create, edit or delete child nodes.

The diagram shown in Figure 8 illustrates the differences in the menus triggered in the same node for users of two different privileges. The user on the left has access to Builder Mode 46 whereby they can edit or delete the particular node as well as create a child node (in this case at the 'Discipline' level) Note also that this user has access to the Clipboard. The user on the right only has access 'read-only' functionality of a normal Mode 48. The options available in each case are shown as option menus 50.

Basic Workflow/Knowledge nodes and their links are stored in two relational database tables. The database management system resides on the server side and can be implemented using any standard RDMS. The first table captures information specific to individual nodes and the second table captures information specific to links between individual nodes. An indicative database schema is shown in Figure 9.

An individual node 30 is an identifiable unit and is stored as an element in a relational database and comprises four fields. The unit is identified by a node

field 50 which provides a unique identifier of the node.
A caption field 52 provides text that can be displayed as
a caption on screen. A description field 54 provides a
longer description which is also displayable. Lastly, a
5 level field 56 is a code which identifies the level of the
node in a hierarchy.

The links between nodes are stored in a second node-node
table with three fields. A from-node field 58 and to-node
field 60 describe the links to and from other nodes and a
10 link-mode field 62 describes the linking mode. Any custom
attributes required for links will also be stored in this
table. The relational database capturing the
Workflow/Knowledge node information is stored on the web
server in the core Knowledge Module 24. Clients are able
15 to view and modify node information from anywhere on the
Internet (or Intranet) and all transactions are managed on
the server so that a high degree of integrity is
maintained if there are multiple concurrent queries.

As previously described, a Workflow structure, (or project
20 roadmap) is a collection of Workflow/Knowledge nodes. A
general description for a Workflow structure is a display
of a collection of Workflow/Knowledge nodes following a
given hierarchical structure where links / dependencies
between nodes exist defining how a node depends on another
25 node. Theoretically, many such structures are possible.
In the domain of project management the following
structures have been defined.

- The Discipline tree structure, where the project is
broken down into a structure that reflects a
30 progression of sub-dividing scientific knowledge
within a particular discipline or area of expertise.
This structure can also be considered as a 'taxonomy'

of a particular area of expertise, such as shown in Figure 3.

- The Integrated tree structure where a project is broken down in a structure that reflects managerial units, i.e. deliverables and sub deliverables of a project. Each node may represent knowledge across disciplines or may represent a single discipline, such as shown in Figure 1.
- A process oriented structure can be represented as a dependency map. For a given Workflow/Knowledge node, dependencies are defined showing how workflow nodes follow each other in the logical sequence of project workflow. For example, the semantics of a particular dependency map may be - "This task cannot be initiated until the completion of these other tasks.", such as shown in Figure 4.

Since any particular project will have different stakeholders, each will require different views onto the system when accessing and creating knowledge. The various dependency maps provide the stakeholder with a mechanism to navigate through the collection of Workflow/Knowledge nodes in a familiar manner reflecting a structure that a specific stakeholder will understand.

The Workflow structure maps will now be defined in more detail.

A tree is a hierarchical structure with a one-to-many relation between a parent Workflow/Knowledge node and children nodes. Thus within the tree structure one (parent) Workflow/Knowledge node has a breakdown in many sub (children) Workflow/Knowledge nodes. These children may in turn have their own children.

A discipline tree reflects a scientific-technical discipline where Workflow/Knowledge nodes are broken down in sub Workflow/Knowledge nodes purely on scientific-technical grounds within a particular discipline. This implies that children nodes (i.e. lower levels in the tree) deal with more detailed knowledge in the same technical-scientific domain as the higher order parent nodes. Thus a specialist with good understanding of a given node should also master the content of high order nodes and parallel nodes in the same discipline tree.

A Multi-discipline or integrated tree reflects a breakdown of the total project deliverables in a tree structured form. There are two primary components to this tree, Project Management Stream (PMS) and Multi-Disciplinary Stream (MDS). PMS breaks down the project into deliverables from a time-management/project flow point of view. MDS breaks down the project in technical project deliverables that straddle all technical scientific disciplines.

For both PMS and MDS, a Workflow/Knowledge node is a portion of the total project deliverable and not specifically the domain of any single technical-scientific specialist. Thus, children Workflow/Knowledge nodes represent various technical-scientific disciplines, with the project deliverable as the common denominator. In other words, the various children of a Workflow/Knowledge node typically represent multiple technical/scientific disciplines. As a user descends to lower levels of these integrated trees, Workflow/Knowledge nodes will begin to duplicate Workflow/Knowledge nodes from the discipline trees since the breakdown starts to become more 'atomic' and representative of the deliverable from an individual technical specialist. Figure 11 shows a discipline tree for a hypothetical discipline (D) and a hypothetical integrated tree (I).

In addition to the tree hierarchy outlined above,
dependencies can be defined between Workflow/Knowledge
nodes of any kind (i.e. nodes from any level in any tree).
Process mode view shows all dependencies for one specific
5 Workflow/Knowledge node, as shown in Figure 4.

The ability to look in different ways at the
Workflow/Knowledge node network is considered as a multi-
dimensional view on the project technical and managerial
knowledge. The different hierarchical tree structures use
10 radically different perspectives to break down the project
to its basic building blocks. The hierarchical tree
structures impose a consistent framework on the project
information. The multi-dimensional aspect (i.e. the
various trees), however, guarantees sufficient flexibility
15 to access the information along different viewpaths or
dimensions.

Links and dependencies are an attribute assigned to a
workflow node which allow it to make connections to other
nodes. Via this mechanism, the workflow structure maps
20 discussed above can be generated. Although both links and
dependencies are used to make connections, they each have
a different role:

- Links are attributes that connect nodes as parents
and children within a hierarchical tree structure.
25 Links are established when creating nodes and the
tree structures. Links are obviously essential to
navigate along the hierarchical trees that form the
basic structure of the system.
- Dependencies are attributes that connect nodes
30 belonging to different trees and allow the dependency
map display. There are two dependency modes:
"dependent on" and "dependent to". For example, node
B is "dependent on" node A and node B is "dependent

to" node C (which is actually equivalent to: node C is "dependent on" node B). It is not possible to have circular dependencies, i.e. it is impossible that node A is "dependent on" node B and that node B is
5 "dependent on" node A. Dependencies can also have different weightings.

Dependencies are created by selecting nodes to be placed on a "node clip-board" while navigating a tree. The user then navigates to the node where the
10 dependencies are to be created, invokes the dependency view and the "drops" the clipboard nodes to the left and right of the dependent node by hitting the left and right arrow on the node in the clipboard.

15 The main function of dependencies is to establish a process view, i.e. to display a map how the information / knowledge of one workflow node is used to feed another workflow node. Another functions of dependencies is to move (navigate) in a logical
20 manner, from a workflow process point of view, between different branches trees or between different trees.

There are a two alternative ways to establish dependencies between workflow nodes:

25 First, Keywords as previously described are assigned to nodes. Keywords are used in a semi-automatic or implicit manner to establish dependencies. In the dependency view of a node all other nodes with a similar keyword set can be displayed and assigned as "dependent to" and "dependent
30 on" the central node. At the same time weighting factors can be assigned to these dependencies.

Second, Visitation and traffic - while the system is in operation during the duration of project, all requests from client to web server are logged and therefore usage metrics can be collected. After a representative period of
5 time, a user can request a view for a given "central" node to display the nodes that have been "visited" by the users that have visited the central node (with some frequency). Subsequently these "also visited" nodes can be displayed and assigned as "dependent to" and "dependent on" the
10 central node. At the same time weighting factors can be assigned to these dependencies.

The objective of the visitation analysis is to suggest connections between nodes that are not yet connected via links (in a tree) or via a dependency.

15 One way this is achieved is as follows. Users of the systems are classified at logon of the system. This classification includes items such as the core discipline of the user and the experience level. At logon the session is also classified. This classification includes items
20 such as "work session", "casual session", "building session". The above criteria are used to developed weighting factors. For example, for a node in a discipline an visit by a highly experienced specialist of another discipline during a "work" session will receive a high
25 weighting factor. Conversely, an visit by an inexperience user during a "casual" session will receive a low weighting factor.

After some period of usage of the system, a query can be conducted to determine which nodes have been visited by
30 each user. Using the weighting factors as described above, points will be assigned depicting the weighted total of visits by a user. A group of nodes for which a logical

connection such as a link or a dependency is to be made is defined as a collection of nodes that score, for a given user, above a given threshold of nodes. When a dependency view / process mode is created for a given node, in the dependencies so created can be displayed including their relative weighting. This relative weighting can be determined in two ways. One by dividing the number of connection points for a given connection by the maximum in the connection group and second by dividing the connection points by the absolute maximum in the system.

Behind each Workflow/Knowledge node, information can be stored. This information is stored in separate documents, on the web server, in a secured document repository document store 20 (Figure 5). The information could range from technical documentation, managerial information such as costs, timing, and effort involved, and managerial hints. These documents can also be simply project documents that need storage (Word Processor documents, spreadsheets emails).

This functionality provides a workflow oriented 'filing cabinet' metaphor whereby a heterogeneous collection of documents can always be accessed in the context of the current state of the project relevant to the particular user.

One important class of information attached to Workflow/Knowledge nodes is basic node documentation. This is contained in a document that is used to store all general information about that specific workflow node. The eXtensible Markup Language (XML) has been used to make it possible to use a single basic, comprehensive document to create a specific display with morphology as demanded by the role of the user.

For example a manager may only wish to see a short synopsis or abstract along with some costing/budgetary information, whereas a technical worker will need a much more detailed view possibly along with crucial journal
5 references.

XML tags are used to identify the portions of basic node documentation. XML (as are other markup languages - most notably HTML) encapsulates information within a starting and an ending tag. In XML the tags are enclosed in angular
10 brackets i.e. <tag>. The ending tag is preceded with a forward slash i.e. </tag>

For example a piece of information could be tagged as follows

```
<TechInfo >  
15     <theory >Piece of information</theory >  
    </TechInfo >
```

The nesting of tags in this instance indicates that the text is intended for a technical reader who wishes to be informed about theory with respect to the subject in the
20 particular Workflow/Knowledge node

```
<ManagInfo>  
    <cost > Piece of information </cost >  
</ManagInfo>
```

This tagging would mark up text intended to inform a
25 manager about costing issues of the subject in the particular Workflow/Knowledge node.

Clearly by tagging textual information in this manner it is easy to extract parts of the document and reassemble customized views for a particular user. This is one of the
30 major benefits of XML.

A comprehensive set of tags has been developed to capture all aspects of any Workflow/Knowledge node. The tagset is specified in a formal document called a Document Type Definition (DTD) against which all XML documents are
5 verified. A 'pseudo-DTD' is outlined below which lists all the available tags for a node description document..

```
<ManagInfo>
<synopsis></synopsis >
<result></result >
10 <cost></cost >
    <recommendation></recommendation >
    <criticalpath></criticalpath >
    <time></time >
    <HR></HR>
15 <teamwork></teamwork >
    <riskmanag></riskmanag >
    <pitfall></pitfall >
    <shortcut></shortcut >
    <QC></QC >
20 <IT></IT >
    <research></research >
    <miscellaneous></miscellaneous >
    <link></link >
    </ManagInfo>
25 <TechInfo >
    <synopsis></synopsis >
    <result></result >
    <overview></overview >
    <theory></theory >
30 <formula></formula >
    <uncertainty></uncertainty >
    <procedure></procedure >
    <input></input >
    <recommendation></recommendation >
35 <cutedge></cutedge >
    <case></case>
```

```
<effort></effort>
<example></example>
<pitfall></pitfall>
<shortcut></shortcut>
5  <QC></QC>
    <IT></IT>
    <research></research>
    <miscellaneous></miscellaneous>
    <TechInfo >
10  <reference>
    <simple></simple>
    <classic></classic>
    <recent></recent>
    <theory></theory>
15  <overview></overview>
    <case></case>
    <techref></techref>
    <internal></internal>
    <miscellaneous></miscellaneous>
20  <link></link>
    </reference>
```

Following is an example of XML marked-up text according to this DTD.

```
<?xml version="1.0"?>
25  <!DOCTYPE nodedoc SYSTEM "M:\XML\DTDs\nodedoc.dtd">
    <nodedoc title="Application of the Archie equation to
derive saturation from wire-line
resistivity logs" author="gb" owner="rtcom">
    <managinfo>
30  <synopsis>Core measured cementation and saturation
exponents (m and n) are
useful when deriving saturation (ie. hydrocarbon content)
from wire-line (resistivity)
logs. </synopsis>
```

<QC>In addition to correct resistivity-porosity measurements, accurate values for cementation exponent (m) and the saturation exponent (n) are essential to arrive at a correct saturation.</QC>

<riskmanag>Error and uncertainty in the parameters m and n are amongst the most significant input parameters that affect uncertainty in saturation, and ultimately, reserve estimates that are sensitive to uncertainty in petrophysical results.</riskmanag>

<pitfall>The derivation of saturation in oil-wet intervals from logs can be roughly estimated but should be considered inadequate as a result of the large variations in the saturation exponent n that would result.</pitfall>

<recommendation>In ideal cases, all necessary parameters can be derived from logs, with exception to the saturation exponent (n). The saturation exponent(s) for a reservoir can only be determined from measurements on cores.</recommendation>

<QC>These measurements take more time and can be costly. Continuous injection techniques performed over the course of at least 14 days eliminate most laboratory effects on resistivity index measurements and are the recommended norm. </QC>

</managinfo>

<reference type="Paper">

<theory>Archie GE (1942); The electrical Resistivity log as an aid in determining some reservoir characteristics. Trans. AIME 146 (January), pp. 54-62.</theory>

</reference>

<techinfo>

<synopsis>The combined wire-line log measurements of
porosity and resistivity, core measurements of cementation
5 and saturation exponents, formation water resistivity, and
clay conductivity provides an empirical approach of
determining well-bore saturation that is completely
independent of physical rock properties such as the
rock capillarity, permeability, and existent
10 buoyancy.</synopsis>
<theory>

There are a number of different approaches available
(Archie, Waxman and Smits, Clavier, Simandoux, Poupon)
many of which are based upon the pioneering
15 principle developed by Archie.</theory>
<procedure>

In the case of clean homogeneous (and water-wet)
sandstone, most of the necessary parameters for the
determination of saturation can be derived from the logs,
20 providing there is a clean fully water-bearing zone (no
residual hydrocarbons!!). Since the ideal conditions of
homogeneous grain texture is rare, some core analysis is
usually appropriate. </procedure>
</techinfo>
25 </nodedoc>

As discussed later when running the system one assumes a
role (manager, technical). Also, for each workflow node
one can display just a synopsis of basic information, a
synopsis ++ (synopsis plus 2 other tagged subjects), or
30 all basic information.

The same mark-up technique is not only used to display
basic node information but is also used to enter new
project information and project documentation. The

facility to reuse and reshape the content is used in this instance to generate final project reports for the various stakeholders. For the purposes of on-the-fly project documentation, an interactive XML editor is being
5 developed that will generate formatted text similar to the example above. The editor will perform all the tagging behind the scenes so that the user may not even know that they are authoring XML.

A further technology is used to translate the XML into a
10 format that can be rendered by standard web browsers. This technology is called XSLT (eXtensible Style Language Transformations) and is being developed in parallel with XML at the World Wide Web Consortium. Using XSLT, the relevant portion of the XML document is transformed (on
15 the server) into HTML by a series of translation templates before it is sent to the users browser.

The mark-up and translation methods described above are not novel. XML and XSLT are rapidly gaining popularity for web based documentation and many customised markup
20 languages are being developed for specific problem domains. However, the combination of XML with the workflow structuring and navigation tools considered novel. Deploying the entire system as a solution on the problems inherent in project management expands on this novelty.
25 The system becomes a tool that can be used to navigate along various structures to a workflow node and subsequently extract information tailored to specific the context of a particular user. As such it becomes a true distributed project management information system as
30 opposed to a substantial amount of unstructured hyper-linked information (as in the format of ordinary HTML web pages).

Collections of Workflow/Knowledge nodes are displayed, navigated and manipulated in a client program called NavMap. NavMap can exist as a standalone program but will be more commonly deployed in a standard web browser for reasons of platform independence and close access to other Internet resources. It uses dynamically generated vector graphics to represent the different project maps and trees, and is optimised to function in a distributed environment. This is an extremely novel interface for project management applications, both for its pure graphical bias and also for the fact that the entire system is totally distributed over the Internet and accessible via low bandwidth connections.

The vector graphics are generated and rendered in a software system developed by Macromedia called Flash. Flash is rapidly becoming a ubiquitous system on the Web. In June 2000, NPD Research conducted a study to determine what percentage of Web browsers have Macromedia Flash preinstalled. The results show that 91.8% of Web users can view Macromedia Flash content without having to download and install a player.

As explained, Macromedia Flash uses vector graphics technology. Unlike bitmapped images that are optimized for a single resolution, vector images can adapt to multiple display sizes and resolutions. Unlike bitmapped images such as GIFs and JPEGs used on the Web today, vector images fit into compact, efficient files that can be transmitted quickly over the Web - even via low bandwidth connections.

Figure 12 shows the NavMap interface and the various options. Note that the browser (IE 5.5) standard toolbars have been turned off and that the view is of a Discipline tree.

The elements of the NavMap are outlined below.

Search Button. This allows a user to rapidly locate a particular node or nodes based on a text string.

5 Home Button. This allows a user to return to the opening screen where they can set preferences and begin a session again.

'Back to Previous' Button. This button allows a user step back in their navigational history to a previous view.

10 'Forward to Next' Button. This button allows a user to return to a view that is forward in their history record. The user must have pressed the 'Back' button for this option to have any effect.

Debugging Button. Displays debugging information.
Developers version only.

15 Refresh Button. Fetches all current information from the server and refreshes the current display.

Discussions Button. Brings the user to a feedback/discussion forum where comments can be posted to the developers. Beta version only.

20 Show Clipboard Button. Shows the content of the clipboard. Used for setting up dependencies in Process Mode.

History Nodes show a record of the nodes that have been traversed and will 'stack up' down the right hand side of the screen as a user session progresses. All nodes in the
25 'history stack' are active and may be clicked upon to take the user to a previous view.

Parent Nodes and Child nodes display in tree views and each are active elements as described in section 2 above.

5 The status line displays expanded information relevant to the node in question. When the cursor is held over a particular Workflow/Knowledge node, an expanded description of the particular node appears in this status line at the base of the screen. When the cursor is held over the same node for more than 2 seconds, a list of the nodes' children replaces the description at the base of
10 the screen.

The remaining Figures 13 to 26 show a number of screen shots and document displays generated from the system:

Figure 13: This is the main opening screen to the NavMap system. A number of user interface 'skins' have been developed so that the aesthetics of
15 this and subsequent screens can be quickly altered. The futuristic look of this screen represents one of these 'skins'.

20 The role of the user can be defined via this screen: a technical role (the wrench button on the left), a managerial role (the brief-case button), and various builder roles (the wrench button left of the brief case, and the flow diagram buttons). The user-role chosen at this
25 point determines the type of displays that are generated when using the system and also the bias of any project documentation that is delivered.

Figure 14: This is the opening screen of the discipline tree (also see section 2.2). The MRS (Multi
30 Disciplinary Reservoir) Project is broken out into four disciplines. The menu which is

shown, appears if the Menu Hotspot belonging to a node is clicked. This menu allows the user to conduct operations required to build the tree (the upper 3 dark-blue shade options), to change tree or display a dependency map (options 4-6, light blue shaded) and to display documents with various degree of detail (options 7-9, no shading). The following sequence of images simulate a user 'drilling-down' into the tree, level by level, with the next image initiated by clicking on the Petrophysics discipline Workflow/Knowledge node (clicking on the left-most node).

Figure 15: The discipline tree is followed into the Well-by-Well Study (clicking on the right-most node).

Figure 16: The discipline tree is followed into the task "Log Derived Lithology, Phi, Sw" clicking on the 3rd node from left).

Figure 17: The discipline tree is followed into the sub-task "Saturation" (right-most node).

Figure 18: The discipline tree is followed into the sub-sub-task " Shaly sand Waxman-Smiths eq.". Below the workflow node the menu options are displayed by clicking on the Menu Hotspot..

Figure 19: Example of managerial synopsis for the "Shaly sand Waxman-Smiths eq." Workflow node (menu option 6). This will be displayed if the system is entered with a managerial role.

Figure 20: Example of all managerial information for the " Shaly sand Waxman-Smits eq." Workflow node (menu option 6). This will be displayed if the system is entered with a managerial role.

5 Figure 21: Example of technical synopsis for the "Shaly sand Waxman-Smits eq." Workflow node (menu option 6). This will be displayed if the system is entered with a technical role.

10 Figure 22: An excerpt from the single XML content file used to generate the various displays shown above in Figures 19, 20 and 21.

15 Figure 23: If, for a given workflow node, the menu option 'Dependency map' is invoked then this dependency display occurs. The map shows the dependencies between the central sub-sub-task " Shaly sand Waxman-Smits eq." and other workflow nodes (studies, tasks, sub-tasks, and sub-sub-tasks) selected from out of the various discipline trees. In the case shown, the central sub-sub-task " Shaly sand Waxman-Smits eq." depends on the four nodes on the left. This means that the results from these nodes are needed as input to the central sub-sub-task " Shaly sand Waxman-Smits eq.". The two nodes on the right depend on the results the central sub-sub-task " Shaly sand Waxman-Smits eq." (in other words, the central node is "dependent to" the nodes on the right.

30 Figure 24: The dependency display allows the user to navigate along the dependency lines. Clicking in the previous display on the upper right workflow node "Net reservoir cut-off criteria" makes that node the central node of a

dependency display. As shown, in dependencies mode the menu shows an option "Tree Mode" to revert back to the tree display.

5 Figure 25: After invoking tree mode, the branch of the
 discipline tree is shown where the workflow
 node is residing that was the last central
 workflow node in dependency mode. In this case
 this is "Net reservoir cut-off criteria" shown
10 second from the right on the second row from
 the bottom.

 Figure 26: This display shows the top level of the Multi-
 discipline or Integrated tree. As discussed
 before, there are two components to this tree,
 Project Management Stream and Multi-
15 disciplinary stream.

Claims

1. A project management system, comprising:
 - a knowledge module arranged to store information relating to a project as a plurality of
5 identifiable units, each identifiable unit being logically linked to one or more other identifiable units;
 - a display module arranged to selectively create a graphical representation of the identifiable
10 units as a tree structure of multiple possible tree structures, the tree structure showing the links between each of the units; and wherein
 - the knowledge module and display module are
15 further arranged to retrieve, on user selection of a given unit, further units being logically linked to the given unit and to create a further graphical representation of those further units.
2. A project management system according to claim 1,
20 wherein each identifiable unit is a node within the system being a portion of a project, the project being defined by the nodes.
3. A project management system according to claim 1 or
25 2, wherein the further graphical representation of the further units is a further tree structure of the multiple possible tree structures.
4. A project management system according to claim 1, 2
30 or 3, the knowledge module being further arranged to store dependency indicators each indicating the precedence order of one identifiable unit in relation to another, the display module being further arranged to create a graphical representation of the

identifiable units in precedence order as indicated by the dependency indicators.

5. A project management system according to any preceding claim, wherein each identifiable unit
5 relates to knowledge within a particular discipline, the knowledge module and display module being arranged to create a graphical representation of the identifiable units as a tree structure for that particular discipline.
- 10 6. A project management system according to any preceding claim, wherein each identifiable unit relates to one of a plurality of subjects, the knowledge module and display module being arranged to
15 create a plurality of graphical representations of the identifiable units as tree structures each tree structure comprising identifiable units for one of the plurality of subjects.
- 20 7. A project management system according to any preceding claim, wherein each identifiable unit relates to one of a plurality of specialists in a technical field subjects, the knowledge module and display module being arranged to create a plurality
25 of graphical representations of the identifiable units as tree structures each tree structure comprising identifiable units relating to one specialist.
8. A project management system according to any preceding claim, wherein the knowledge module
30 comprises a database in which the identifiable units are stored.

9. A project management system according to claim 8,
wherein each identifiable unit includes a unique
identifier and descriptive text.
- 5 10. A project management system according to any
preceding claim, wherein the knowledge module
includes a link store comprising logical links
between identifiable units.
- 10 11. A project management system according to any
preceding claim, wherein the knowledge module
includes a connection module for creating logical
connections between identifiable units.
- 15 12. A project management system according to claim 11,
wherein the connection module is arranged to create
logical connections as a function of the graphical
representations selected by a user.
- 20 13. A project management system according to claim 11 or
12, wherein the connection module is arranged to
create logical connections as a function of the
changes in graphical representations selected by a
user.
- 25 14. A project management system according to claim 11, 12
or 13, wherein the logical connections are links.
15. A project management system according any preceding
claim, further comprising a document store arranged
to store a plurality of documents, each being
associated with one of the plurality of identifiable
units.

16. A method of creating a graphical representation of a project in a project management system, comprising:
- storing information relating to a project as a plurality of identifiable units, each identifiable unit being logically linked to one or more other identifiable units;
 - creating a graphical representation of the identifiable units as a tree structure of multiple possible tree structures, the tree structure showing the links between each of the units;
 - retrieving, on user selection of a given unit, further units being logically linked to the given unit and to create a further graphical representation of those further units.
17. A method according to claim 16, wherein each identifiable unit is a node within the system being a portion of a project, the project being defined by the nodes.
18. A method according to claim 16 or 17, wherein the further graphical representation of the further units is a further tree structure of the multiple possible tree structures.
19. A method according to claim 16, 17 or 18, further comprising storing dependency indicators each indicating the precedence order of one identifiable unit in relation to another, and creating a graphical representation of the identifiable units in precedence order as indicated by the dependency indicators.
20. A method according to any of claims 16 to 19, wherein each identifiable unit relates to knowledge within a

particular discipline, and comprising creating a graphical representation of the identifiable units as a tree structure for that particular discipline.

21. A method according to any of claims 16 to 20, wherein
5 each identifiable unit relates to one of a plurality of subjects, and comprising creating a plurality of graphical representations of the identifiable units as tree structures each tree structure comprising identifiable units for one of the plurality of
10 subjects.
22. A method according to any claims 16 to 21, wherein
each identifiable unit relates to one of a plurality of specialists in a technical field subjects, and comprising creating a plurality of graphical
15 representations of the identifiable units as tree structures each tree structure comprising identifiable units relating to one specialist.
23. A method according to any of claims 16 to 22,
20 comprising storing the identifiable units in a database in a knowledge module.
24. A method according to any of claims 16 to 23, wherein
each identifiable unit includes a unique identifier and descriptive text.
25. A method according to any of claims 16 to 24,
25 comprising storing logical links between identifiable units in a link store.
26. A method according to any of claims 16 to 25, further
comprising creating logical connections between identifiable units.

27. A method according to claim 26, comprising creating logical connections as a function of the graphical representations selected by a user.
- 5 28. A method according to claim 26 or 27, comprising creating logical connections as a function of the changes in graphical representations selected by a user.
29. A method according to claim 26, 27 or 28, wherein the logical connections are links.
- 10 30. A computer program arranged to carry out the steps of any of claims 16 to 29.
31. A computer based system for providing a graphical representation of a project in a project management system from a server computer to a client computer, comprising:
- 15
- means for storing information relating to a project as a plurality of identifiable units, each identifiable unit being logically linked to one or more other identifiable units;

20

 - means for creating a graphical representation of the identifiable units as a tree structure of multiple possible tree structures, the tree structure showing the links between each of the units;

25

 - means for retrieving, on user selection of a given unit, further units being logically linked to the given unit and to create a further graphical representation of those further units.
32. A computer based system according to claim 31, wherein each identifiable unit is a node within the
- 30

structure comprising identifiable units relating to one specialist.

38. A computer based system according to any of claims 31 to 37, comprising means for storing the identifiable units in a database in a knowledge module.
39. A computer based system according to any of claims 31 to 38, wherein each identifiable unit includes a unique identifier and descriptive text.
40. A computer based system according to any of claims 31 to 39, comprising means for storing logical links between identifiable units in a link store.
41. A computer based system according to any of claims 31 to 40, further comprising means for creating logical connections between identifiable units.
42. A computer based system according to claim 41, comprising means for creating logical connections as a function of the graphical representations selected by a user.
43. A computer based system according to claim 41 or 42, comprising means for creating logical connections as a function of the changes in graphical representations selected by a user.
44. A computer based system according to claim 41, 42 or 43, wherein the logical connections are links.



INVESTOR IN PEOPLE

Application No: GB 0023952.5
Claims searched: 1 to 44

Examiner: John Donaldson
Date of search: 22 March 2002

Patents Act 1977 Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:
UK CI (Ed.T): G4A(AUB, AADB, AUXF, AUXX)
Int CI (Ed.7): G06F 17/00, 17/30, 17/60
Other: Online:WPI, EPODOC, JAPIO

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
X	GB 2313933 A (MATHEWS), see abstract, page 11, line 9 to page 18, line 16	1 to 44
X	EP 0314596 A2 (IBM), see abstract, column 6, line 30 to column 7, line 3, column 11, lines 7 to 27	1 to 44
X	EP 0314594 A2 (IBM), see abstract, column 5, lines 29 to 47	1 to 44
X	WO 00/30000 A2 (TRIPORT TECHNOLOGIES), see abstract	1 to 44
X	WO 00/07125 A1 (SCHAWK), see abstract	1 to 44

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.